ENGINEERING

LEVEL

UNIVERSITY OF SOUTHERN CALIFORNIA

SEMIANNUAL TECHNICAL REPORT

Ramakant Nevatia
Alexander A. Sawchuk
Project Directors

Covering Research Activity During the Period
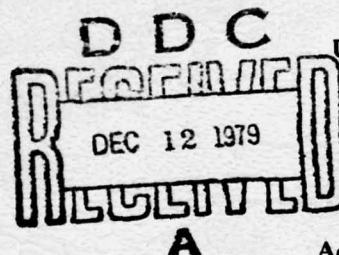1 April 1979 through 30 September 1979

30 September 1979

Image Processing Institute
University of Southern California
University Park
Los Angeles, California 90007

IPI

IMAGE PROCESSING INSTITUTE
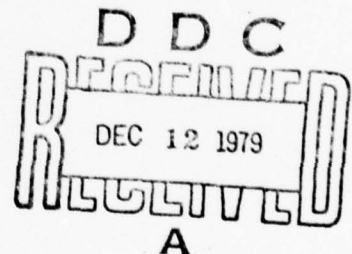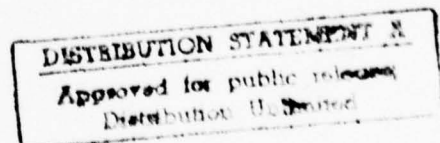
SEMIANNUAL TECHNICAL REPORT

Covering Research Activity During the Period
1 April 1979 through 30 September 1979

Ramakant Nevatia
Alexander A. Sawchuk
Project directors
(213) 741-5514

D D C

RECEIVED

DEC 12 1979

A

Image Processing Institute
University of Southern California
University Park
Los Angeles, California 90007

30 September 1979

79 12 11 001

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
USCIPI Report-910		

4. TITLE *(and Subtitle)*	5. TYPE OF REPORT & PERIOD COVERED
IMAGE UNDERSTANDING RESEARCH.	Semi-annual Technical 1 April 1979-30 Sept 1979
	6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s)
Ramakant Nevatia (Project Directors) Alexander A. Sawchuk	F-33615-76-C-1203

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Image Processing Institute University of Southern California University Park, Los Angeles, Calif.90007	ARPA Order No. 3119

11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE
Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209	September 30, 1979
	13. NUMBER OF PAGES
	154

14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*
Wright Patterson Air Force Base U.S. Air Force Air Force Avionics Laboratory Air Force Systems Command, Ohio 45433	UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for release: distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Key Words: Digital Image Processing, Image Restoration
Scene Analysis, Image Understanding,
Edge Detection, Image Segmentation, CCD Arrays,
CCD Processors.

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This technical report summarizes the image understanding and smart sensor research activities performed by the USC Image Processing Institute during the period of 1 April 1979 through 30 September 1979 under contract number F-33615-76-C-1203 with the Advanced Research Projects Agency, Information Processing Techniques Officer, and monitored by the

DD FORM 1 JAN 73 **1473**   EDITION OF 1 NOV 65 IS OBSOLETE

Wright-Patterson Air Force Base, Dayton, Ohio.

The research program has, as its primary purpose, the development of techniques and systems for understanding images. Methodologies range from low level image processing principles, smart sensor CCD LSI circuit design, up to higher level symbolic representations and relational structure manipulations.

ABSTRACT

This technical report summarizes the image understanding and smart sensor research activities performed by the USC Image Processing Institute during the period of 1 April 1979 through 30 September 1979 under contract number F-33615-76-C-1203 with the Advanced Research Projects Agency, Information Processing Techniques Office, and monitored by the Wright-Patterson Air Force Base, Dayton, Ohio.

The research program has, as its primary purpose, the development of techniques and systems for understanding images. Methodologies range from low level image processing principles, smart sensor CCD LSI circuit design, up to higher level symbolic representations and relational structure manipulations.

PROJECT PARTICIPANTS

Project Directors   Affiliation

Ramakant Nevatia   Computer Science & Electrical
                                      Engineering
Alexander A. Sawchuk   Electrical Engineering

Research Staff   Affiliation

Jean-Francois Abramatic   I.R.I.A. - FRANCE
  (Visiting Scientist)
Keith E. Price   Image Processing Institute
William K. Pratt   Electrical Engineering

Support Staff

Ted Bleecker   Charles McManis
Gary Edwards   Mary Monson
John Horner   Ray Schmidt
Eileen Jurak   Thomas Tertocha
Hilda Marti   Amy Yiu
Toyone Mayeda

Students

Behnam Ashjari

K. Ramesh Babu

Bir Bhanu

Peter Chuan

David Garber

Kenneth I. Laws

Sang Uk Lee

Chun Moo Lo

Felicia Vilnrotter

TABLE OF CONTENTS

1. <u>Research</u> <u>Overview</u>

This document presents results of research over the past six months at the USC Image Processing Institute. This report is divided in two major areas: image understanding projects, and smart sensor projects. These areas are abstracted below.

<u>Image</u> <u>Understanding</u> <u>Projects</u>

The image understanding research has been directed at the various levels of an image understanding system. In the first section, Abramatic and Lee describe further developments in techniques of rapid convolution of images using small generating kernels.

A major area of research in the last six months has been texture analysis and this report contains four sections related to this topic. Laws describes texture energy measures using small (3x3, 5x5) filter masks. The classification results appear to be better than for the more expensive gray level co-occurence computation methods. Vilnrotter, Nevatia and Price present new resuls in automatically generating structural texture descriptions for natural textures. Garber describes a mathematical model for textures and its use in texture analysis as well as synthesis. Ashjari present a theoretical analysis of the probability density function of the singular value of a random texture field. These singular values are believed to be useful texture discriminants.

In work at higher levels of our image understanding system, Babu describes an approach to the recognition of structured objects such as airports. Price presents new results in matching map models to aerial images of Stockton Ca. and San Diego, Ca. areas.

Finally, Lo and Sawchuk present a comparison between a maximum <u>a</u> <u>posteriori</u> (MAP) non-linear filter developed by them and a conventional linear minimum-mean-square error filter for restoration

1

of images degraded by Poisson noise.

Smart Sensor Projects

The Hughes Research Laboratories present a section describing research progress on the development of smart sensors for image processing. Hughes has completed the construction of a new CCD chip and is currently testing and evaluating its performance. This chip is designed to perform the following functions:

3x3 Laplacian

5x5 median filter

5x5 programmable weight convolver

7x7 bipolar convolver

26x26 edge detection convolver.

2. Image Understanding Projects

2.1 Parallel-Cascade   Convolution   of   Images: Fixed-Point
Implementation

J.F. Abramatic and Sang Uk Lee

Introduction

Convolution of images has proved to be very useful in many areas
including image enhancement, image restoration and edge detection.
However, whenever the size of convolution kernel is large, the
processing can be very costly in time as well as in memory
requirements. Among the recent techniques for alleviating this
problem is the idea of sequentially convolving the image with small
size kernels [1]. Our work in SVD/SGK filter follows this new
approach.

SVD/SGK Convolution Filters

Using Singular Value Decomposition (SVD) technique, we can
decompose any LxL impulse response $\underline{H}$ as the sum of R separable 2-D
filters [2].

$$\underline{H} = \sum_{i=1}^{R} \psi_i \underline{a}_i \underline{b}_i^t \qquad (1)$$

where R is the rank of H, ψ_i , i=1,...R are the singular values of $\underline{H}$,
which are the square roots of the eigenvalues of $\underline{H}^T\underline{H}$, $\{a_i\}$,$\{b_i\}$ are
the eigenvalues of $\underline{H}^T\underline{H}$ and $\underline{H}\underline{H}^T$, respectively. Whenever R is less than
L, the number of operations needed to perform the convolution can be

3

reduced. Also, when ψ_i becomes small, the decomposition can be truncated without introducing significant errors in the overall filtering process.

In general, we will be satisfied with a multistage expansion that will closely approximate $\underline{H}$. In this case

$$\underline{H} \cong \sum_{i=1}^{K} \psi_i \underline{a}_i \underline{b}_i^t$$

where K is the number of retained terms ($K < R$). In many cases, the impulse response $\underline{H}$ can be approximated with a small error by only retaining the first few dominant terms of Eq. (1). This suggests a parallel implementation of the 2-D filter described in Fig. 1. Moreover, each of the filters H_i, where $H_i = \psi_i \underline{a}_i \underline{b}_i^t$, can be realized by cascading 3x1 SGK filters, oriented either in the column or in the row direction. Figure 2 shows the basic implementation structure of SVD/SGK convolution filter.

Fixed-point Finite-Word Length Arithmetic Implementation of SVD/SGK Filters

To take advantage of the SVD/SGK filters, one will often have to implement them in special-purpose hardware. Image processing display is one example where such filters may be used. They will perform convolution without needing the capabilities of a general purpose host computer. In such case, the use of fixed-point finite-word length arithmetic will be required if one wants to have "real-time" capabilities. ["Real-time" will mean here to be able to perform convolutions between two frame displays. The unit of time will basically be 1/30 second]. Using fixed-point arithmetic will introduce errors in the computations. We can classify these errors into three types: a) filter coefficients quantization error, b) roundoff error, c) overflow error. The quantization errors occur because the coefficients of the filters will not take the exact value
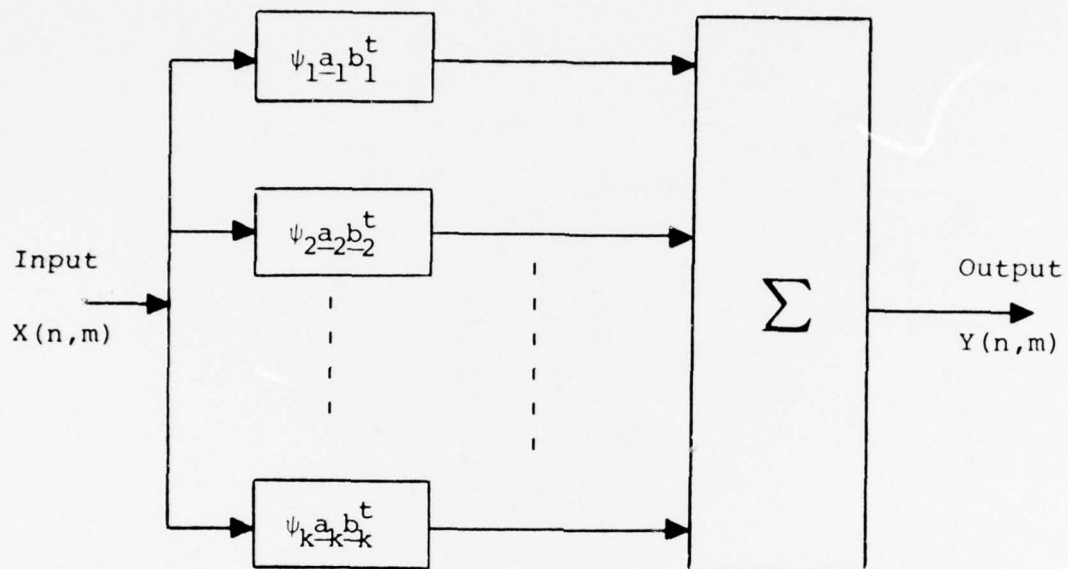
4

Figure 1. Block diagram showing implementation
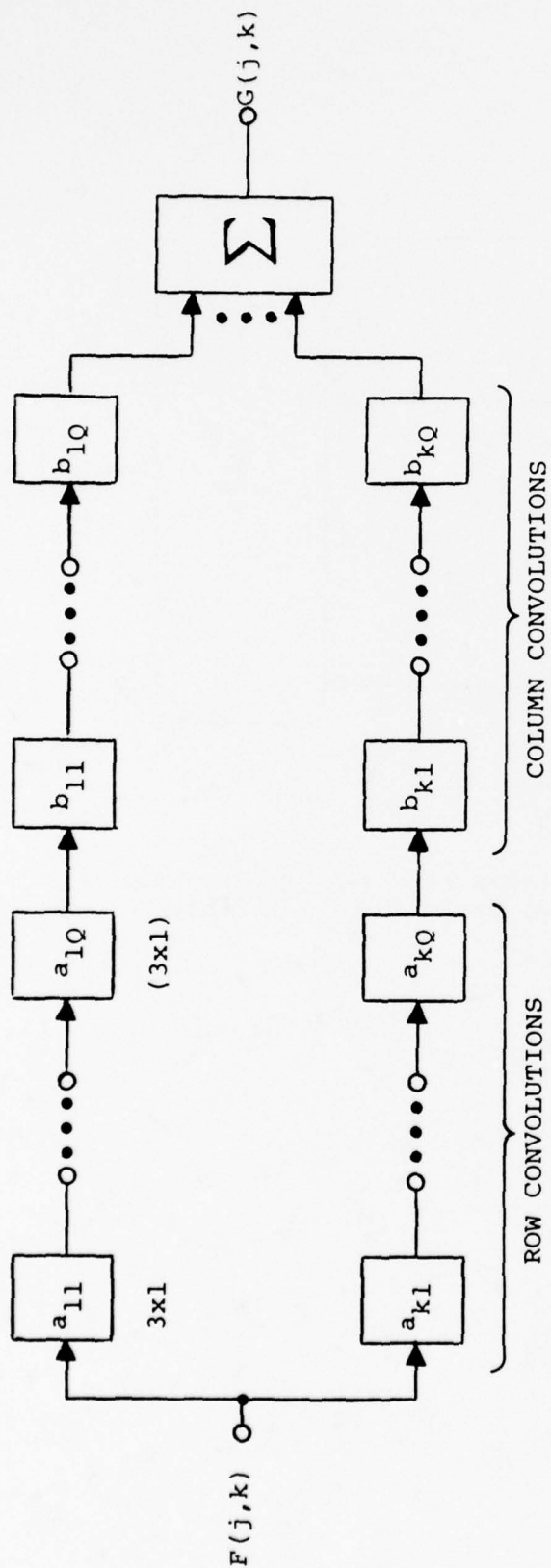of k-stage separable 2-D filter.

Figure 2. SVD/SGK Convolution Filter Structure.

they were assigned by the design procedure. This is due to the finite word length used to represent these numbers. Experimental results have proved that the amount of errors of this type is not significant in our case. Roundoff errors occur every time a multiplication is performed. Multiplying an n bits (including sign bit) number by an m bits number will result in an n+m-1 bits number. Generally, only p bits (p < n+m-1) will be available to store the result of multiplication. Rounding/truncation operation will be performed introducing the so-called roundoff error. Finally, the use of fixed point arithmetic will constrain the numbers to belong to a given dynamic range say -1 to +1. Performing an addition between two numbers may result in a number which does not belong to this interval. The truncation operation that will assign the limit value to the result (say -1 or +1) will introduce a so-called overflow error.

To minimize the discussed errors, we have two kinds of degrees of freedom: a) every subfilter of the cascade structure can be scaled by any factor as long as the overall gain of the filter is kept to its actual value, b) the order with which the elements of the cascade are used, can be chosen as we wish. We will have thus to derive a scaling procedure that will deal with the overflow problem and an ordering procedure for minimizing the effect of roundoff errors.

Scaling Procedure

Let us consider the case where a $(2Q+1) \times (2Q+1)$ filter has been decomposed by means of the SVD/SGK structure. We will be concerned here by one SVD expansion that implements the cascade convolution of Q 1-D column operation filters and Q 1-D row operation filters, all of them of size 3 x 1. The only information we have on the input is that it has been normalized to the range [0,1]. We will use the sum-scaling procedure to prevent overflow errors. (See Gold-Rabiner [3]). If $h^i(k,\ell)$ denotes the impulse response of the system from the input to the ith subfilter output, $h^i(k,\ell)$ is given by

$$h^i(k,\ell) = S_i [g^1(k,\ell) ** g^2(k,\ell) ** \ldots ** g^i(k,\ell)] \tag{2}$$

where $g^i(k,\ell)$ is defined to be

7

$$
g^i(k,\ell) = \begin{cases} \begin{bmatrix} 0 & C_1 & 0 \\ 0 & C_2 & 0 \\ 0 & C_3 & 0 \end{bmatrix} & \text{if } i^{th} \text{ section is column} \\ & \text{oriented filter} \\[2em] \begin{bmatrix} 0 & 0 & 0 \\ r_1 & r_2 & r_3 \\ 0 & 0 & 0 \end{bmatrix} & \text{if } i^{th} \text{ section is row} \\ & \text{oriented filter} \end{cases} \tag{3}
$$

S_i is a scaling factor. Using the sum-scaling procedure, we will insure that no overflow occurs by imposing

$$
\sum_k \sum_\ell |h^i(k,\ell)| = 1 \qquad \text{for } i = 1,2,\ldots,2Q \tag{4}
$$

As each of the subfilter filters are either row or column operation subfilter, $h^i(k,\ell)$ can be rewritten as

$$
h^i(k,\ell) = S_i g_C^i(k) g_R^i(\ell) \tag{5}
$$

where $g_C^i(k)$ and $g_R^i(\ell)$ are the impulse responses obtained by gathering the j column and the (i-j) row operation subfilter, respectively. Equation (4) can then be rewritten as

$$
S_i = \frac{1}{\displaystyle\sum_k |g_C^i(k)| \sum_\ell |g_R^i(\ell)|} \tag{6}
$$

The scaling factor used for the ith section is finally given by

---

**denotes the 2-D convolution while * denotes the 1-D convolution.

$$\sigma_i = \frac{S_i}{S_{i-1}} \tag{7}$$

One can see then that this factor will be the same as the one that one would have used cascading the j column or the (i-j) rows only. To prove that, let us consider that the ith section is a column oriented filter. In that case σ_i is given by

$$\sigma_i = \frac{\sum\limits_k |g_C^{i-1}(k)|}{\sum\limits_k |g_C^i(k)|} \tag{8}$$

The terms corresponding to the row convolutions in S_i and S_{i-1} will be identical. This result will lighten the burden of calculating the scaling factors as the overall 2-D scaling procedure reduces to two 1-D scaling procedures. We will denote $\gamma^i(k,\ell)$ the scaled impulse response of the ith section.

Modelling of the Round-off Errors

When an n+m-1 bits number is rounded to an p bits number, the error introduced can be assumed to be the realization of a random variable uniformly distributed between $-2^{-(P+1)}$ and $2^{-(P+1)}$. A more restrictive assumption will be to consider that all the roundoff errors occurring as a result of rounding/truncation are independent random quantities. Figure 3 describes one stage SVD expanded filter roundoff noise model

Assuming that the errors are independent, one can then easily calculate the variance of the overall roundoff error

$$\varepsilon = \sigma_R^2 \sum_{i=1}^{2Q} \left(\sum_k \sum_\ell |f(k,\ell)|^2 \right) \tag{9}$$
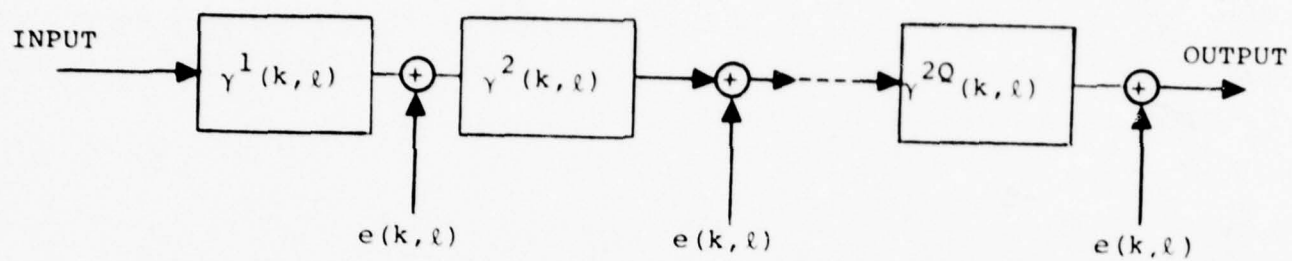
9

Figure 3.   SVD/SGK Filter Roundoff Noise Model.
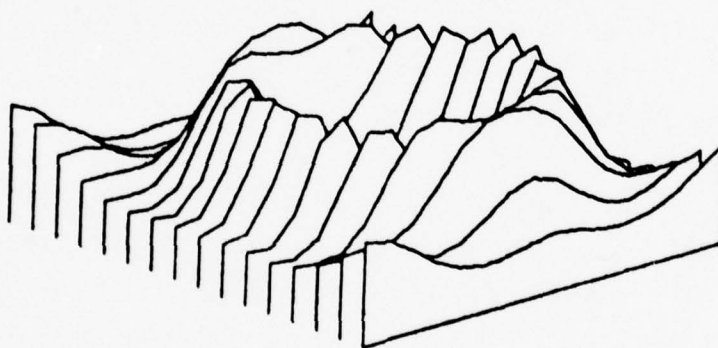

Figure 4.   Prospective View of Prototype Filter
Frequency Response.

10

where

$$\sigma_k^2 = \frac{2^{-2P}}{12} \tag{10}$$

and

$$f_i(k, \ell) = \gamma_{(k, \ell)}^{i+1} ** \ldots ** \gamma_{(k, \ell)}^{2Q} \tag{11}$$

$f_i(k, \ell)$ is a separable impulse response and thus the error can be rewritten as

$$\varepsilon = \sigma_R^2 \sum_{i=1}^{2Q} \left(\sum_k |f_C^i(k)|^2 \sum_\ell |f_R^i(\ell)|^2 \right) \tag{12}$$

This formulation will be useful to generalize to the 2-D separable case the ordering algorithm proposed by Chan and Rabiner [4].

Ordering Procedure

Dealing with a $(2Q+1) \times (2Q+1)$ kernel, gives $2Q$ subfilters of size of 3×1 either column or row operators. We thus have $(2Q)!$ possible orderings. As in the 1-D case testing all the possibilities will become almost impossible as Q grows, but good sub-optimal procedures will prove to be efficient. This is the reason why we will generalize the 2-D separable case with the Chan-Rabiner's 1-D sub-optimal algorithm.

This algorithm consists in building the cascade structure section after section starting from the end, minimizing at each step the new term introduced in Eq. (12). Following this approach, we will show that the resulting ordering of the rows (resp. the columns) will be the same as the one obtained by ordering the rows (resp. the columns)

11

as if we were solving two 1-D problems. The algorithm will provide then a good way of ordering the rows "among" the columns. To show that result, let us assume that we have already chosen Q-i sections, j of them are columns, Q-i-j are rows. We want to determine the ith section of the cascade structure. Let us define C_i, R_i as

$$C_i = \sum_k |f_C^{i+1}(k)|^2 \tag{13}$$

$$R_i = \sum_\ell |f_R^{i+1}(\ell)|^2 \tag{14}$$

If we choose a column as the ith section, the new term in Eq. (12) will be

$$E_i^C = R_i \sum_k |f_C^i(k)|^2 \tag{15}$$

if it is a row we will have

$$E_i^R = C_i \sum_\ell |f_r^i(\ell)|^2 \tag{16}$$

It is clear now that to get the smallest E_i, one has only to compare the results given by the row and the column that would have been chosen in the 1-D algorithm orderings.

Ordering Algorithm

The Ordering algorithm for one 2-D separable channel goes as follows.

   i) Run the 1-D Chan-Rabiner ordering algorithm for the rows and for the columns separately. (The calculation of the scaling factors is included in this step.)

   ii) For i going from 2Q to 1, compare E_i^C and E_i^R given by Eqs. (15), (16) to decide whether a row or a column should become the ith section.

This algorithm has been tested by using as inputs either 2-D white noise or real images. The results have been compared to other "common sense" orderings. In all cases the ordering provided by the algorithm has proved to be the best.

Experiments

We have chosen to use a zero-phase bandpass filter with a size of 11x11 as a prototype filter. The frequency response of the prototype filter is shown in Fig. 4. The prototype filter was expanded using the SVD technique and approximated by retaining only the 4 largest terms in Eq. (1). The resulting approximating filter yields 0.92% normalized mean square error (NMSE). Several experiments have been performed to evaluate the SVD/SGK convolution filter with fixed-point arithmetic. The original image is shown in Fig. 5a and the result of convolution with a 11x11 prototype filter with a floating point arithmetic is shown in Fig. 5b. The result of convolution with SVD/SGK filter with a fixed-point arithmetic is presented in Fig. 5c. In this experiment, we assigned 12 bits (including sign bit) word-length for memory and 16 bits for filter coefficient quantization. The resulting NMSE between the prototype processed output of Fig. 5b and the SVD/SGK processed output with fixed-point
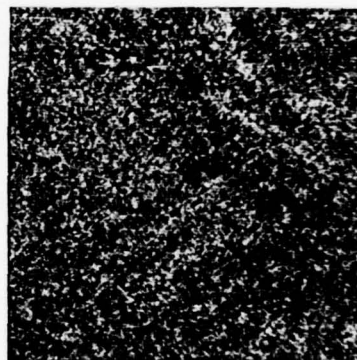
13
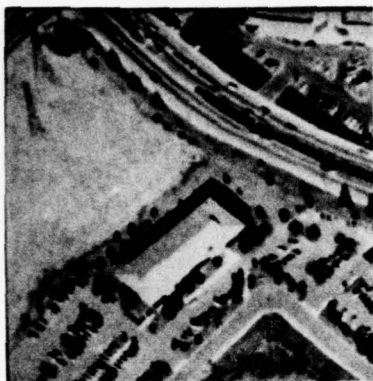
(a)   Original
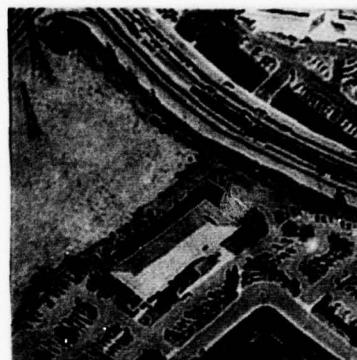
(b)   Direct

(c)   SVD/SGK Convolution

(d)   Absolute difference x 150

Figure 5.   Example of Image Convolution with Prototype
Filter and SVD/SGK Filter

14

arithmetic is 0.8742%. We show also the magnitude of absolute difference between Figs. 5b and 5c as guide to perceive the errors. The difference image of Fig. 5d was multiplied by scaling factor 150. To give a practical idea of how useful is the SVD expansion of $\underline{H}$, Figs. 6 and 7 show the convolution output and difference image corresponding to different degree of approximation. The usefulness of SVD expansion can be demonstrated by noting that we can tradeoff between the amount of approximation error and the computational efficiency by selecting the number of terms in the SVD expansion of $\underline{H}$.

Summary

The singular value decomposition of a 2-D finite impulse response has proved to be useful in designing 2-D approximating FIR filters that can be implemented by means of a parallel-cascade structure. We have shown how the algorithms available in the domain of 1-D signal processing for implementing cascade filters with fixed point finite-word length arithmetic, can be extended to 2-D signal processing. It has been seen that taking advantage of the 2-D structure by interlacing column and row convolutions sequentially, can be useful in minimizing the roundoff errors. Finally, experiments on real images have proved that the SVD/SGK filters can be useful in the domain of "real-time" image convolution.

References

1. W.K. Pratt, J.F. Abramatic and Sang Uk Lee, Two-Dimensional Small Generating kernel Convolution," University of Southern California, Image Processing Institute, USCIPI Report 860, March 1979, pp. 76-85.

2. S. Treitel and J.L. Shanks, "The Design of Multistage Separable Planar Filters," IEEE Trans. on Geosci. Electron, Vol. GE-9, No. 1, Jan. 1971, pp. 10-27.

3. L.R. Rabiner and B. Gold, Theory and Application of Digital Signal
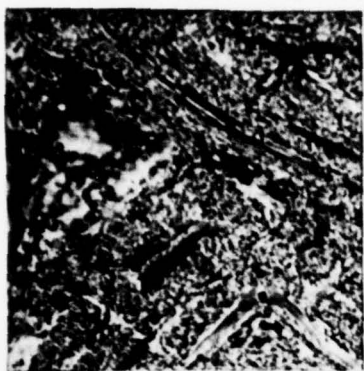
(a)   1 term SVD/SGk

(b)   2 term SVD/SGK

(c)   3 term SVD/SGK

Figure 6.   Example of Image Convolution Using SVD/SGK Filter
with Different Degree of Approximation of $\underline{H}$

16

(a)   Scale factor = 10

(b)   Scale factor = 30

(c)   Scale factor = 32

Figure 7.   Absolute Difference Image Between Fig. 5b and Fig. 6

Processing, Prentice-Hall, New Jersey, 1975.

4. D.S.K. Chan and L.R. Rabiner, "An Algorithm for Minimizing Roundoff Noise in Cascade Realization of Finite Impulse Response Digital Filters," Bell Sys. Tech. J., Vol. 52, No. 3, March 1973, pp. 347-385.

2.2   Texture Energy Measures

Kenneth I.   Laws

This paper presents a set of "texture energy" transforms that provide texture measures for each pixel of a monochrome image. The transforms are fast, requiring only one-dimensional convolutions and simple moving-average techniques. The method provides more accurate classification than gray level co-occurrence methods. It is local, operating on small image windows in much the same manner as the human visual system. It can be made invariant to changes in luminance, contrast, and rotation without histogram equalization or other preprocessing.

Figure 1 shows the sequence of images, or image blocks, used in measuring texture. The original image is first filtered with a set of small convolution masks with fixed, integer coefficients. These masks are separable, so that only one-dimensional convolution is required. The filtering can also be accomplished with two-stage 3x3 or 1x3 and 3x1 convolutions. It could even be done optically.

The filtered images are then processed with a nonlinear "local texture energy" filter. This is simply a moving-window average of the

Original   Filtered   Energy   Principle
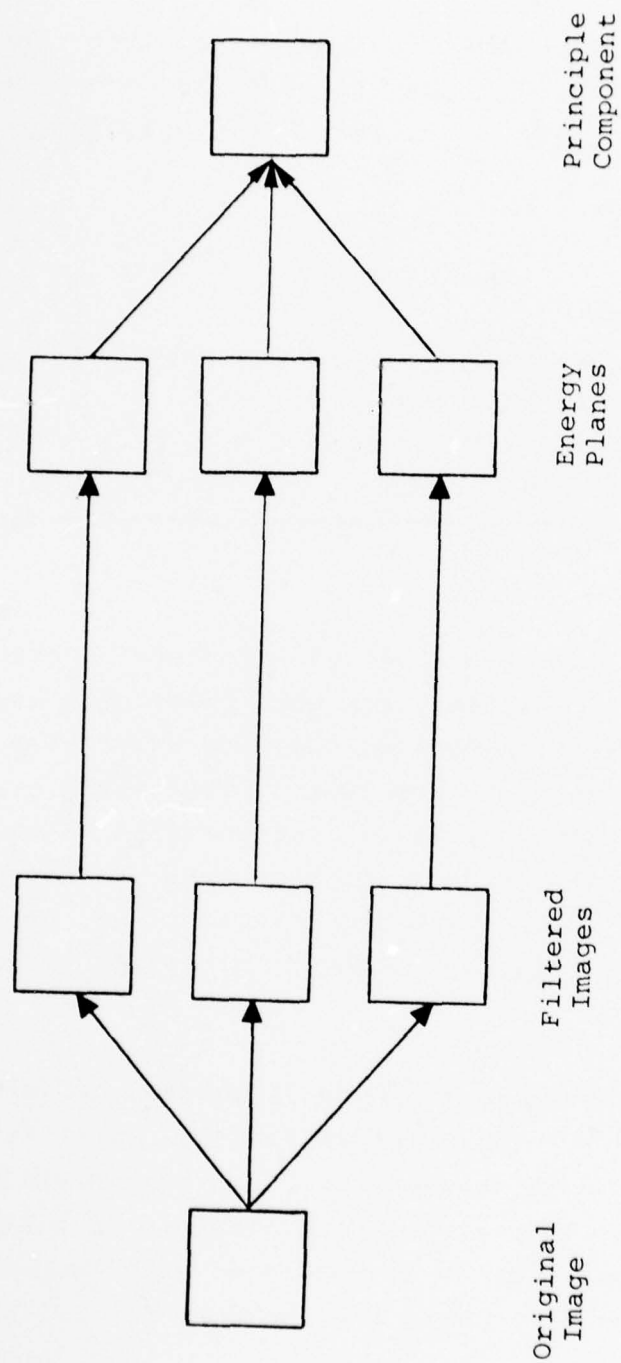Image   Images   Planes   Component

Figure 1. Texture Measurement.

absolute image values. Such moving-window operations are very fast even on general purpose digital computers. Each pixel is examined only twice, making the transform time independent of window size. The best window size depends on the size of image texture regions. This study has concentrated on 15x15 windows. Even smaller windows could be used if color information were available. The number of rows which must be processed at one time is equal to the window size.

Figure 1 shows a one-to-one mapping between filtered images and texture energy planes. In previous research we used twelve measures, based primarily on the four statistical moments suggested by Faugeras and Pratt. Experience has shown that either variance or standard deviation alone is sufficient to extract texture information from the filtered images.

Variance is an average squared deviation from the mean. For a zero mean field, as produced by convolution with a zero-sum mask, variance is the average of squared signal values. It is thus an energy measure, in the formal sense of the word. Macro-window variance measures the total energy within a window. If the image has been filtered, it measures the local energy within the pass band. The SDV macro-statistic is the square root of this local energy. It may be considered a "texture energy" measure. A faster energy transform, the ABSAVE statistic, will be introduced shortly.

These statistics differ from previously studied frequency-domain texture measures because of their local nature. Frequency components are measured with very small convolution masks. Each micro window is treated independently, without regard to its phase relationships with other micro windows. This method, similar to human visual processing, is appropriate for textures with a short coherence length or correlation distance. It is less powerful than Fourier methods for man-made textures with inherent synchronization of texture element spacings.

The final step in Figure 1 shows the linear combination of the texture energy planes into a smaller number of principle component planes, typically three. This is an optional data compression step. It is tempting to call the final images "perceptual planes," but it has not yet been proven that they relate to human texture perception. They do seem to represent natural texture dimensions, and to be more "reliable" than the texture energy planes.

The final output can also be a segmented image. A classifier assigning texture labels to the image pixels can take either texture energy planes or principle component planes as input. Classification is simple and fast if texture classes are known a priori. Cluster or segmentation algorithms must be used if texture classes are unknown.

Texture Data

In an experimental study, the results can be no better than the input data. We require a set of uniform texture fields large enough to provide adequate samples of each texture. Ideally this training set should come from a target application area. For a general vision system, each texture must be a "natural" one, and the set must include a range of natural texture dimensions. We avoid artificially generated textures, such as sinusoidal gratings, because they would favor the Fourier transform and other frequency domain measures.

The textures we have chosen are from high-resolution photographs used in the Brodatz texture album. Figure 2a shows a composite. The first two rows of 128x128 blocks are from the images of Grass, Raffia, Sand, Wool, Pigskin, Leather, Water, and Wood. The lower-left quadrant is composed of 32x32 blocks, and the lower-right quadrant of 16x16 blocks. The 128x128 blocks have been individually histogram equalized, the other blocks have been equalized by quadrant. Histogram equalization has been used to remove all first-order differences. This also finesses the problem of whether to measure image luminance or density, since the equalization gives the same
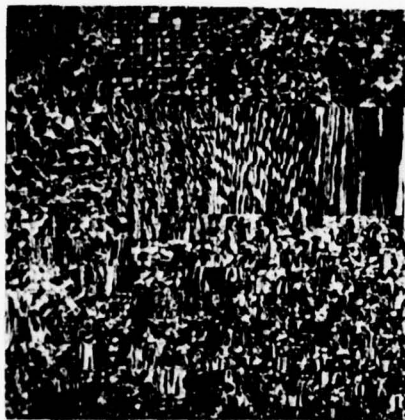
(a) Composite
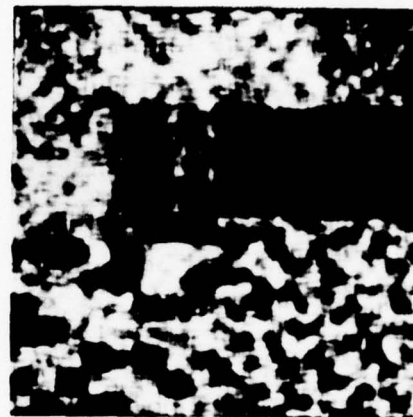
(b) First Component

(c) Second Component

(d) Classification

Figure 2.   Texture Images

result for either. In our discriminant analyses a more rigorous adaptive equalization has also been used: it seems to give more conservative and reliable results.

The textures were chosen precisely because they are difficult to discriminate. They are a worst case dataset. Grass and Sand are very similar, with the main difference being the extended edges in grass. Pigskin, Raffia, and Sand may be considered cellular textures with similar cell sizes. Raffia is distinguished by its long-range structure, and Wool by its fiber content and lack of coarse edge structure. The Grass, Leather, Wood, and Water images all have vertical structure.

Comparison Statistics

Co-occurrence matrices are a popular source of texture features. We have generated co-occurrence matrices from 15x15 source windows requantized to 32 gray levels. Each matrix is thus 32x32. Nine of these matrices are used, corresponding to horizontal and vertical spacings of zero, one, and seven pixels. The chosen spacings correspond to horizontal, vertical, and top left to bottom right diagonal directions. The P00 matrix records first-order information: all the entries are on the diagonal. The other eight matrices record second-order information. The matrices are not symmetric, nor is there any averaging across different co-occurrence angles.

Table 1 shows the classification accuracies available with various feature sets. The first analysis uses only the ASM, CON, COR, IDM, and ENT Haralick moments [1], [2]. Together the 32 features give almost 68% classification accuracy on the adaptively equalized texture set. The globally equalized textures generate two dominant discriminant functions using P10CON, P01IDM, P70IDM, P11CON, P01CON, P10IDM, P10COR, and P11COR. Discriminant functions for adaptively equalized textures use P10CON, P01IDM, P70CON, P11CON, P01CON, and P71COR. The angular second moment, correlation, and entropy features

23

apparently carry little texture information.

Table 1. Co-occurrence Classification Accuracy

Feature Set	Global	Adaptive
Haralick Moments	70.85	67.58
Rectilinear Moments	63.04	65.92
Diagonal Moments	56.60	63.04
Combined Moments	72.07	68.16

The second and third analyses in Table 1 use rectilinear and diagonal moments, respectively. These moments will not be described here. Neither set is as powerful as the Haralick moments. The fourth analysis combines all of the co-occurrence features, a total of 172 independent texture measures for the nine co-occurrence matrices. Classification accuracy improves very little, and the variables selected by the discriminant analysis are nearly all from the Haralick set.

Macro-Statistic Selection

Table 2 shows the classification accuracies possible with local texture energy measures. The classifier used 50 feature planes per texture, each produced by convolution with a 3-vector, 5-vector, 3x3 matrix, or 5x5 matrix. The SDV statistics were gathered over 15x15 macro windows. Classification accuracies are much higher than the 72% achieved with co-occurrence statistics.

Energy and variance are both defined as sums of squares because such sums are analytically tractable. The physical world is under no constraint to be tractable. It is probable that the human visual

24

system avoids root-mean-square computations, and quite possible that simpler statistics are more appropriate for texture analysis.

Table 2. Macro-Statistic Classification Accuracies

Feature Set	Global	Adaptive
SDV	85.99	85.60
ABSAVE	88.09	87.11
SDV+ABSAVE	89.16	87.55

Table 2 also presents an alternative to the standard deviation. The ABSAVE statistic is computed as the average absolute value within a macro window. For a zero mean field it may be considered a fast approximation to the standard deviation. It performs poorly only with operators which are not zero-sum. For this dataset, ABSAVE features are jointly more powerful than SDV features, and nearly as powerful as both sets together.

The SDV and ABSAVE macro-statistics share a common weakness. Neither can distinguish between a dark field with bright spots and a bright field with dark spots. In statistical terms, the two fields differ in skewness. In frequency terms, they differ in phase rather than energy. One solution is to average positive values instead of absolute values. It is reasonable that neurons in the retina might perform such a clipping function. Such one-sided measures perform slightly less well than the SDV and ABSAVE measures, although much better than co-occurrence statistics. Another possibility is the measurement of local phase relationships, as with moving-window Fourier transforms. For the present we accept the ABSAVE statistic, keeping in mind that there will be some textures we cannot

discriminate. ABSAVE features are preferred to SDV features only because of their computational simplicity. Both appear to be equivalent measures of texture energy.

Center-Weighted Filter Masks

Figure 3 shows three sets of one-dimensional convolution masks. The names are mnemonics for Level, Edge, Spot, Wave, Ripple, Undulation, and Oscillation. The vectors in each set are ordered by sequency. The vectors are weighted toward the center, all are symmetric or anti-symmetric, and all but the Level vectors are zero-sum. The vectors in each set are independent, but not orthogonal.

The 1x3 vectors form a basis for the larger vector sets. Each 1x5 vector may be generated by convolving two 1x3 vectors. S5, for instance, can be generated as L3(*)S3, S3(*)L3, or E3(*)E3. The 1x7 vectors can be generated by convolving 1x3 and 1x5 vectors, or by twice convolving 1x3 vectors. The sequency of a generated vector is the sum of the component sequencies.

We have investigated the discriminant power of one-dimensional masks. Rotation-invariant filters, such as the Sobel gradient magnitude, are only fair as texture measures. Better results are obtained by using directional masks separately and then combining the texture energy measures. We have applied horizontal and vertical masks in pairs, although the discriminant analyses have not been constrained to assign equal weights.

The six 3-vectors alone perform slightly better than the elaborate co-occurrence features. This is amazing considering the simplicity of the texture energy method and the many experimental vindications of Haralick's co-occurrence statistics. The 5-vector statistics perform even better, achieving 82% classification accuracy. Using 7-vectors or combining more than one vector size gives no

```
L3 =    1    2    1
E3 =   -1    0    1
S3 =   -1    2   -1


L5 =    1    4    6    4    1
E5 =   -1   -2    0    2    1
S5 =   -1    0    2    0   -1
W5 =   -1    2    0   -2    1
R5 =    1   -4    6   -4    1


L7 =    1    6   15   20   15    6    1
E7 =   -1   -4   -5    0    5    4    1
S7 =   -1   -2    1    4    1   -2   -1
W7 =   -1    0    3    0   -3    0    1
R7 =    1   -2   -1    4   -1   -2    1
O7 =   -1    6  -15   20  -15    6   -1
```

Figure 3. Center-Weighted Vector Masks


```
 1   2   1          -1   0   1          -1   2  -1
 2   4   2          -2   0   2          -2   4  -2
 1   2   1          -1   0   1          -1   2  -1
    L3L3                L3E3                L3S3


-1   2  -1           1   0  -1           1  -2   1
 0   0   0           0   0   0           0   0   0
 1   2   1          -1   0   1          -1   2  -1
    E3L3                E3E3                E3S3


-1  -2  -1           1   0  -1           1  -2   1
 2   4   2          -2   0   2          -2   4  -2
-1  -2  -1           1   0  -1           1  -2   1
    S3L3                S3E3                S3S3
```

Figure 4. 3x3 Center-Weighted Masks

27

significant improvement.

Neurological studies show that the visual cortex computes edge measures in approximately ten-degree increments. We have investigated diagonal one-dimensional features, although they are not properly members of the separable feature sets. Inclusion of diagonal features improves classification accuracies significantly. The 5-vector statistics alone are sufficient to achieve 86% classification accuracy, close to the maximum reached in this study. The combined feature sets have little more power, but provide insight into the selection process. The discriminant functions are based on vectors of all directions and sizes. Different subsets are selected in the globally equalized and adaptively equalized cases. Yet all of the selected features are either Edge statistics or the symmetric Spot, Ripple, and Oscillation statistics. None of the anti-symmetric Wave or Undulation features were found useful.

Figure 4 shows the nine masks generated by convolving a vertical 3-vector with a horizontal 3-vector. This may be considered a cross-product or vector multiplication operation, but convolution has special significance here. We shall extract texture information from image data by convolving with the 3x3 masks. Convolution with the component one-dimensional masks gives exactly the same result as convolution with the separable 3x3 mask.

The nine independent 3x3 masks form a complete set. Any 3x3 matrix can be expressed as a unique linear combination of the masks. The 5x5 masks and 7x7 masks also form complete sets, with even stronger weighting toward the center. The separable structure of these masks makes it feasible to apply them as spatial-domain filters. A 5x5 convolution, for instance, can be implemented as two 3x3 convolutions, a 5x1 and a 1x5 convolution, or two 3x1 and two 1x3 convolutions.

The two-dimensional masks are even more powerful than the tested sets of one-dimensional masks. Again the length five masks are best, although the evidence is less conclusive. Classification accuracies are in the range 86% to 88%. The adaptively equalized 3x3+5x5 feature

28

subset differs from the 5x5 feature subset only by the inclusion of L3S3, the ninth and last feature to be added. Analyses with 7x7 masks have shown no significant improvement. Selected statistics again differ from one analysis to another, but Wave features are rare and Undulation features are absent. The consistent inclusion of R5R5 is somewhat surprising since matching image structures must be quite rare. This mask resembles a two-dimensional sinc or Bessel function. The similar S5S5 feature is individually very strong, but has little power when combined with other features. Apparently it measures the same texture energy as R5R5.

Combining one-dimensional and two-dimensional features improves classification accuracy very little. Two-dimensional features enter the models first, followed by a few of the longer vector features. Again there are few Wave and no Undulation features, despite the fact they are individually strong discriminating features. Otherwise the selection seems somewhat arbitrary.

Two sets of filter masks have been found which work well: separable square masks and rotated vector masks. Very likely the human visual system uses rotated circular masks. For digital image processing the square masks are the most convenient. This dataset requires only the Level, Edge, Spot, and Ripple 5x5 masks. Classification of 15x15 blocks can be done with accuracies above 86% using just the Level, Edge, and Ripple or the Level, Spot, and Ripple subsets.

Classification Results

The Level (or L5L5) texture energy transform is sensitive to changes in luminance level. The moving-window average, or AVE statistic, can be used as a brightness measure for segmentation purposes. Standard deviation, or SDV, can be used as a local contrast measure. The other filters are inherently insensitive to low frequency luminance changes, and can be made invariant to contrast

changes by taking the ratio of ABSAVE values to the L5L5 SDV values. This normalization reduces classification accuracy with the zero-sum filters by about two percentage points. The contrast-invariant features may be used to segment or classify image textures without prior histogram equalization. Averaging of ABSAVE values from rotated filter masks, such as L5E5 and E5L5, can be used as rotation-invariant texture measures. The usefulness of such measures depends on the application area. In a general vision system it is better to use directional measures, and to allow a higher level processor to decide which textures are equivalent.

We have applied the contrast-invariant transforms to the composite texture image. Figure 2 shows the first two principle component planes, scaled and histogram equalized. The third plane (not shown) is similar to the second with contrast in the first quadrant reversed. The discriminant dimensions are the same ones found with co-occurrence features and with every other texture set we have tried.

Figure 2d shows the results of classifying every pixel into one of the eight texture categories. Coefficients of the classification functions were derived from 1156 samples per texture, or approximately 1/250 of the possible 15x15 sample windows. Twelve texture transforms from the 5x5 Level, Edge, Spot, and Ripple subset were used. Processing time was about 25 minutes on a PDP KL/10.

It can be seen that the large blocks of Wool, Water, and Wood are almost perfectly classified. Grass is the most poorly classified, with only 25% accuracy. Most of the misclassified Grass pixels are labeled Leather, although other labels are also common. Sand and Pigskin are also confused. The 32x32 blocks are very well separated, although it may be difficult to tell in this gray-scale picture. Even the 16x16 blocks are differentiated to an extent. Visual appearance could be enhanced by leaving "doubtful" pixels unclassified.

References

[1] R.M. Haralick, K. Shanmugam, and I. Dinstein, Textural Features
for Image Classification," IEEE Trans. on Syst., Man, and Cybern.,
vol. SMC-3, pp. 610-621, Nov. 1973.

[2] J.S. Weszka, C.R. Dyer and A. Rosenfeld, "A Comparative Study of
Texture Measures for Terrain Classification," IEEE Trans. on Syst.,
Man, and Cybern., vol. SMC-6, pp. 269-285, April 1976.

2.3   Automatic Generation of Natural Texture Descriptions

F. Vilnrotter, R. Nevatia and K. Price

Introduction

Previously [1,2] we reported on some initial experiments with a
method for generating descriptions of natural textures. This report
will review the basic method and present the actual description system
in more detail. A historical discussion is presented in [1 and 2] and
is not included in this report.

Our goal in texture description is to automatically generate
descriptions of texture patterns which are similar to those generated
by people and can also be used as a part of the symbolic description
of objects in a scene. As a first step in this description system we
have looked at the problem of describing regular texture patterns.
After some initial experimentation we settled on studying how edge
elements repeat in the texture area.   Edges are less affected by
variations caused by noise and small intensity changes than the
original image value.

31

The basic operation on an edge image is to compute an edge
repetition array (essentially the binary case of a gray level
co-occurrence matrix). These array values are counts of the number of
times an edge occurs at a point and at another point at a given
distance and direction. We use 4 directions with the distance ranging
from 2 to 32 (more in some cases). Several variations on the basic
computation are required, involving the direction of the edge (the
edge detector produces edges quantized to 8 directions). When looking
in the horizontal direction the only edges which would correctly
indicate the boundary of a texture element would be edges in a
vertical direction. Correspondingly in other scan directions only the
perpendicular edge directions are used. Other restrictions on
allowable directions can produce useful results. Consider a texture
field composed only of vertical stripes. Each stripe would have an
edge on each side, one going up, one down. If the arrays are computed
using only those pairs with opposite directions (up and down) then
there should be large values in the repetition array at distances
equal to the size of the stripes (to distinguish intensity of the
stripes, down with up is separate from up with down - later referred
to as the dark and light totals). Additionally, if exactly the same
directions are required then the spacing of pairs of elements should
have large values in the repetition arrays. Finally the totals are
normalized based on the total number of edges in the window in the
given directions

Example

A raffia subwindow and edge image along with their corresponding
element spacing and element size graphs are given in Figure 1. Figure
1 c and d each contain 4 graphs, 1 for each of the 4 scan directions:
horizontal, 45 degrees, vertical, and 135 degrees. The solid line
represents dark element information while the broken line graph
represents light element information. The most significant results in
our example occur along the vertical scan direction; there is a sharp
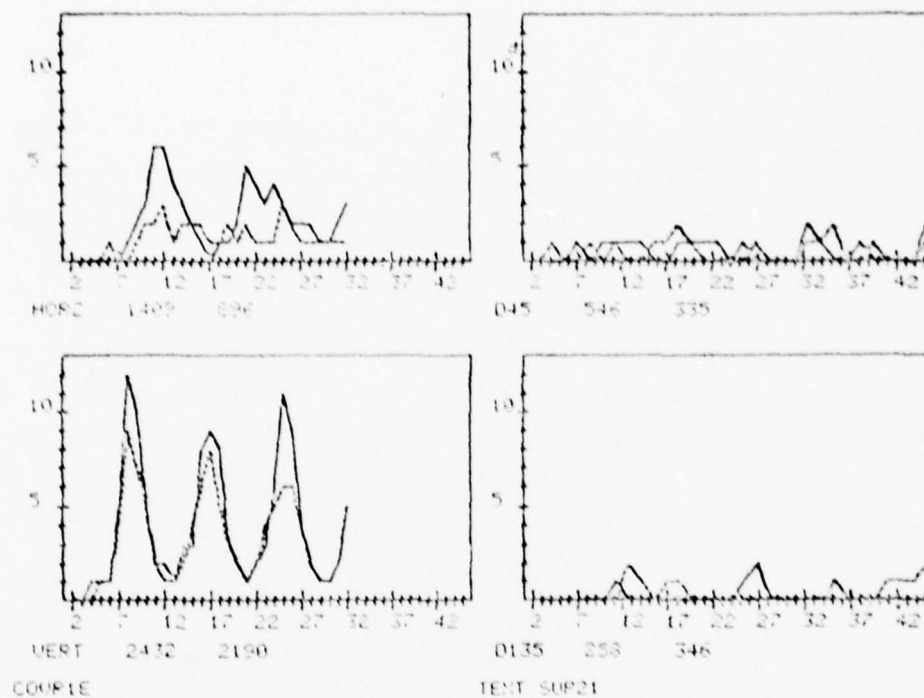peak at 8 and its approximate multiples for both light and dark
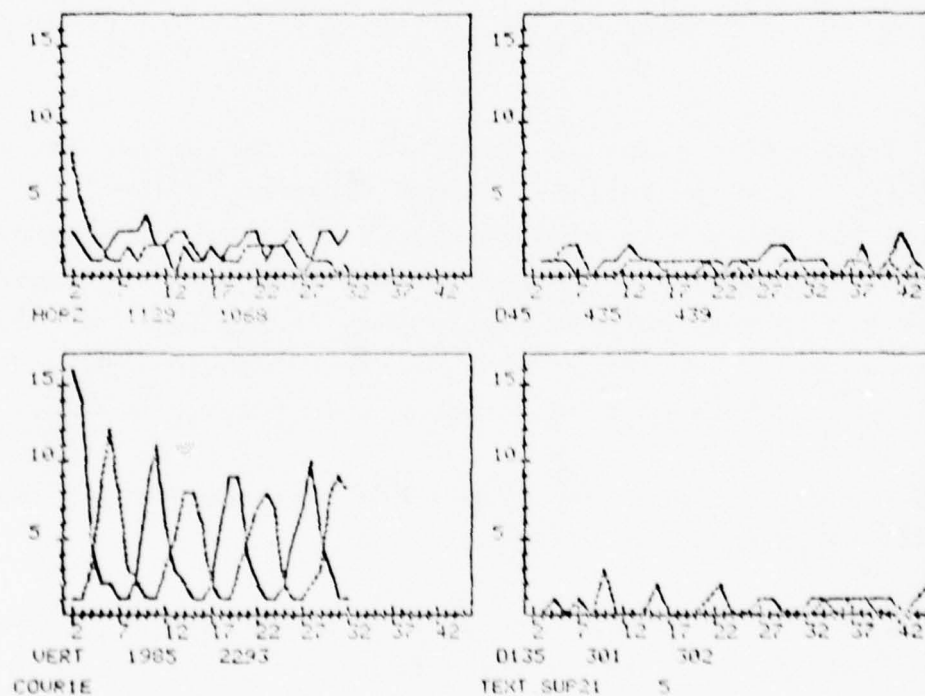
32
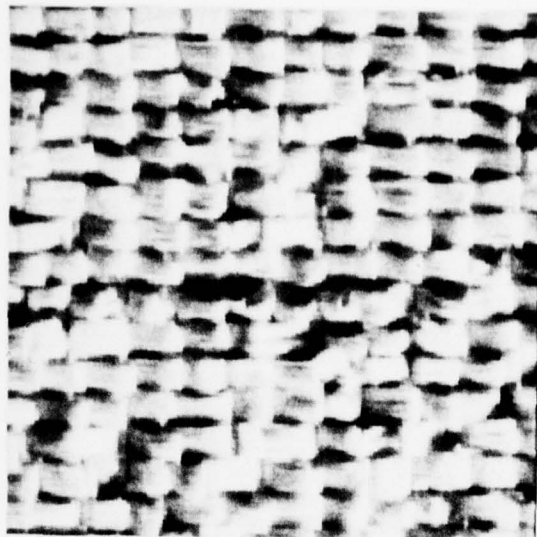
a)   Raffia Subwindow
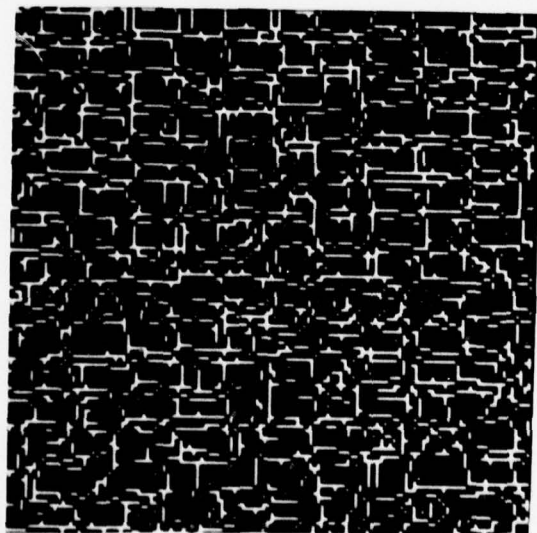
b)   Non-maximal Suppressed Edges from a

Fig. 1.   RAFFIA.TST

33

c) Raffia Element Spacing Graphs

d) Raffia Element Size Graphs

Fig. 1. CONTINUED

34

objects in the vertical direction element spacing graph. The vertical element size graph shows peaks at 2 and 6 repeating at intervals of 8. One could conclude that dark objects of element size 2 and light objects of element size 6 repeat at intervals of 8.

The program (to be discussed in detail below) calculates edge repetition arrays from the edge and direction images of a texture subwindow. It then interprets these results noting evidence of periodicity, element spacing and/or predominant element size, and their relative strength. These descriptions are generated for both dark and light objects in each of 4 scan directions.

The methodology of the program is summarized below:

1). The element spacing graph is examined to determine if there are any element spacing peaks which repeat, i.e., if there are major or minor peaks (as defined later) at multiples of a given element spacing value.

2). The element size graph is examined to determine if there is support for any potential element spacing value. Support is established for element spacing value, N, if there are moderate to high peaks at element size values: M, M+N, M+2N,... for some element size value M. If such support exists then it can be said that there is evidence of periodicity with element spacing N; and there is evidence that elements of size M repeat with spacing N.

3). If periodicity cannot be established then the element size graph is searched for predominant element sizes.

The remainder of this report will be a descriptive outline of the program's 2nd part. This program is not in its final form, and all results are preliminary.

The major program sections are as follows:

1). Generation of Edge Repetition Arrays - The description of this section was provided in a previous report [1].

2). Construction of PEAK-VALLEY SIGNATURE - A peak-valley signature is constructed for each of the graphs. It has 1's in places where peaks occur, -1's in places where Valleys occur and 0's in places which are hillsides - for example, the vertical direction, solid line graph in Fig. 1c has signature -1, 0, 0, 0, 0, 0, 1, 0, 0, 0, -1, 0, 0, 0, 1 for values 2 thru 16. Later in the program, peaks are further classified as major, minor or subminor and the 1's in the signature are adjusted to reflect this classification by being set to 3, 2, or 1 respectively. This signature is used to provide easy access to location, width and classification of peaks.

3). Determine amplitude descriptors and constant profile - The amplitude of each point on the graph is classified as Very Strong, Strong, Moderate, Weak or Very Weak. These classifications are assigned according to amplitude ranges, e.g. a point on the graph which has amplitude greater than 15 would be assigned VERY STRONG as its amplitude descriptor, a point whose amplitude lies between 10 and 15 would be assigned STRONG, etc. These descriptors are stored in the amplitude descriptor matrix. The constant profile of a graph is a descriptor which categorizes a graph in terms of how closely it approximates a constant. Both of the above are used in the final (description generation) section.

The constant profile of the vertical broken line graph in figure 1c is 0 since this graph does not approximate a constant. However, the 135 degrees broken line graph has a constant profile of 4 since it approximates a constant fairly well.

4). Collect Peak Information - The following information is catalogued for each peak for use later in the program:

a). location of left valley,

b). location of right valley,

c). difference in amplitude between peak and left valley,

d). difference in amplitude between peak and right valley,

e). peak width,

f). valley to valley width,

g). amplitude,

A list of peaks is also formed in this section to facilitate the sequential examination of peaks. Peaks with maximum peak-valley variation (the maximum of c and d above) greater than a predetermined floor value are labelled minor peaks (later some of these peaks may be determined to be major peaks). All others are subminor peaks.

5). Extract Major Peaks - A list of major peaks is formed in this section. A peak is considered to be major if:

a). its amplitude is above a predefined floor value,

b). MIN left peak-valley amplitude difference, right peak-valley amplitude difference is above a predefined floor value, and

c). its amplitude is above a predetermined percentage of the maximum graph value.

In Fig. 1c there is only one broken line graph having major peaks. It is the vertical direction broken line graph and its major peaks occur at 8, 17, 25 and 26.

6). Determine repetition descriptors - In the first part of this section the element spacing graph is used while the second part uses element size information.

a). Use of element spacing graph - Peaks are classified as major, minor and subminor. Subminor peaks are too weak to be used in this section.

37

The first major peak is considered as a possible element spacing value. Checks are made to determine if peaks occur at multiples of this value. Then the appropriate descriptor is assigned as follows:

3 - This descriptor is assigned if each multiple of the value being considered is the location of a major peak. (At least one repetition must be found.).

2 - This descriptor is assigned if at least one multiple of the value being considered is the location of a major peak and the rest are at least minor peaks. (At least 1 repetition must be found).

1 - Same as for 2 with the exception that one of the repetitive peaks is missing. In this case at least 1 repetitions must be found.

0 - none of the above cases apply.

If the descriptor is greater than zero all repetitions of the peak presently being considered are tagged. The next peak to be considered is the next untagged major peak.

The process continues until all untagged major peaks are categorized.

b). Use of element size graph - The element size graph is used to lend support to the evidence found in a.

Element spacing values are considered in order of decreasing positive repetition descriptor value, say D1, determined in Ia. If it is observed that peaks occur at locations M, M+N, M+2N,... for element size M and element spacing value N then the size repetition descriptor, say D2, for N is set to 3, 2, 1 or 0 as explained in a.

NOTE: When repetitions are being sought a margin of 1.5 to the left and right is given.

If the descriptor produced while considering element size M is greater than zero repetitions of the peak namely M+N, M+2N,... are both temporarily and permanently tagged. For a given N, D2 is maximized over element size peaks which are not temporarily tagged. After all element size peaks which are greater than subminor peaks and which are not temporarily tagged are considered the temporary tags are reset and the next element spacing value is considered.

If the descriptor produced while considering element size M is greater than zero then element size, M, is assigned an ordered pair, say <S1,S2>, where S1 is the spacing value N at which the size value M repeats and S2 is the repetition descriptor currently produced. S1 and S2 are reset only if the current descriptor produced is greater than the current value of S2, i.e., S2 is maximized over element spacing values with positive D1.

The repetition descriptors D1 and D2 associated with untagged element spacing peaks as well as the ordered pairs <S1,S2> associated with non-permanently tagged element size peaks are considered when descriptions are being generated in the interpretation section described below.

For the set of graphs in Figs. 1c and d the vertical element spacing value 8 has D1=D2=3 for both light and dark objects. The vertical element size values 2 (for dark objects) and 6 (for bright objects) have <S1,S2>=<8,3>

7). Interpretation Section - A simplified outline of this section is given in Fig. 2. If all of the case statements and nested conditionals were expanded this section would essentially consist of a set of production rules of the form:

    IF(COND1 AND COND2 AND...AND CONDN)
    THEN PRINT("APPROPRIATE DESCRIPTIVE MESSAGE").
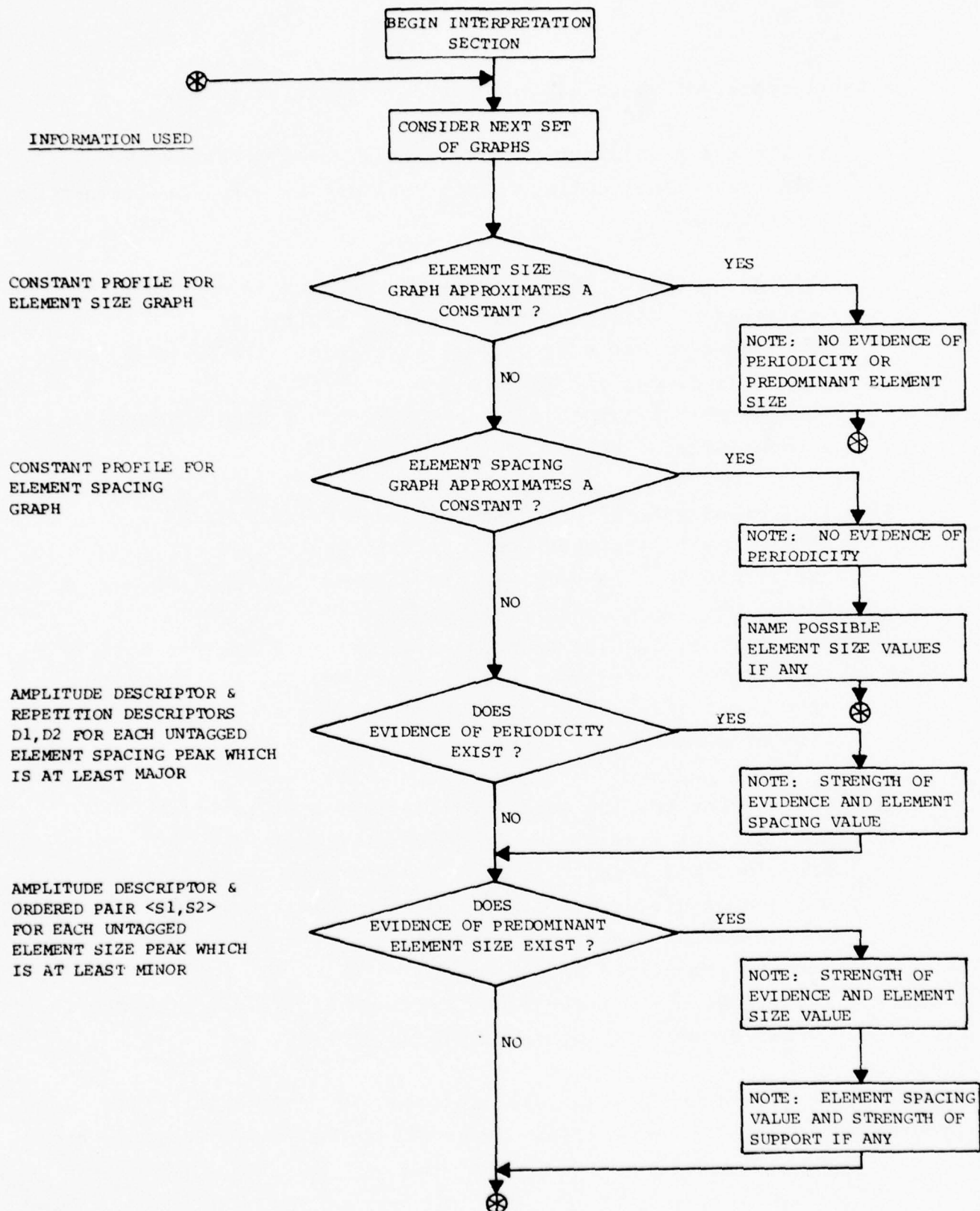
Fig. 2. Texture Interpretation Algorithm

A small subset of these expanded production rules follows:

Rule 1: IF(Element size graph closely approximates a constant)
THEN PRINT("No evidence of periodicity or predominant element size").

Rule 2: IF(Element spacing graph closely approximates a constant)
AND (Element size graph has constant profile 0)
AND (There exists a major peak at element size value N with amplitude descriptor STRONG)
THEN PRINT("No evidence of periodicity. STRONG evidence of element size of N").

Rule 3: IF(Element spacing graph has constant profile 0)
AND (Element size graph has constant profile 0)
AND (There is a major peak at element spacing value, N, having amplitude descriptor STRONG)
AND (Element Spacing repetition descriptor, D1, for N is 3)
AND (Element size repetition descriptor, D2, for N is 3)
THEN PRINT ("There is VERY STRONG evidence of periodicity with element spacing N").

Rule 4: IF (Element spacing graph has constant profile 0)
AND (Element size graph has constant profile 0)
AND (There is a major peak at element size value N)
AND (Amplitude descriptor for N is at least STRONG)
AND (N is not permanently tagged)
AND (S2 associated with N is 3)
THEN PRINT ("There is STRONG evidence of element size N with STRONG support for element spacing S1").

The constant profile of a graph is defined in the description of program section 3. Repetition descriptors D1, D2, (S1, S2) are

41

defined in the description of program section 6. Tagging is also discussed in that section. Since the program is not in final form use of the rules listed above is tentative.

The description generated for the set of graphs given in figure 1 is given in figure 3.

Results

The mosaic pictured in figure 4a contains 2 sample images of raffia in the top row and 2 sample images of herringbone material in the 2nd row. Figure 4b contains the corresponding edge images for these samples. The highly regular patterns of raffia and herringbone material produces strong periodic patterns in the plots of the edge repetition arrays given in Figures 5-8 a and b.

The most significant results for raffia are seen in the vertical scan direction. Major peaks repeat at multiples of 8 or 9 in the vertical element spacing graphs (Figs. 5a, 6a). Likewise peaks at 2 (for dark objects) and 6 (for light objects) repeat at increments of 8 in the vertical element size graphs (Figs. 5b, 6b). This would tend to indicate that dark elements of size 2 repeat with a spacing of 8; and that light elements of size 6 repeat with a spacing of 8. The automatic descriptions for raffia given in Figures 5c, 6c report similar evidence. This is consistent with the results for the sample of raffia discussed earlier (see Figures 1, 3). Less significant is the periodic result found in the horizontal scan direction for dark objects. Dark objects with widths of roughly between 7 and 9 repeat with element spacing 11 (Fig. 5a,b and 6a,b). The horizontal dark element graphs lend strong support to regularity. However, the amplitude of the peaks in the element size graphs is so low that element size evidence is considered weak (Figs. 5,6b and c). The only other evidence worth noting in the horizontal scan direction is the evidence of element size 2 for light objects. No evidence of periodicity is found for light objects.

42

FILENAME = RAFFIA.TST

DARK OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

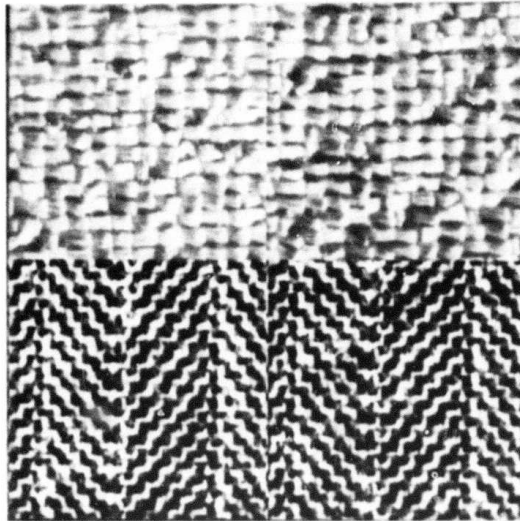   THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 11.00000

   THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 12.00000

   THERE IS WEAK EVIDENCE OF ELEMENT SIZE 10.00000
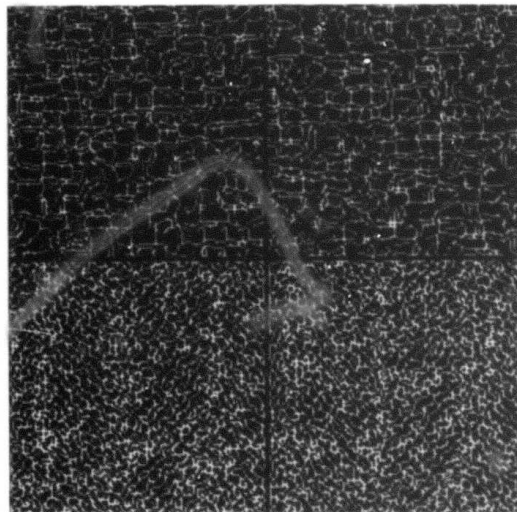   WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

45 DEGREE SCAN DIRECTION

   NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

   THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 8.000000

   THERE IS STRONG EVIDENCE OF ELEMENT SIZE 2.000000
   WITH STRONG SUPPORT FOR ELEMENT SPACING 8.000000

135 DEGREE SCAN DIRECTION

   NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

LIGHT OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

   NO EVIDENCE OF PERIODICITY
   MODERATE EVIDENCE OF ELEMENT SIZE OF 2.000000

45 DEGREE SCAN DIRECTION

   NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE
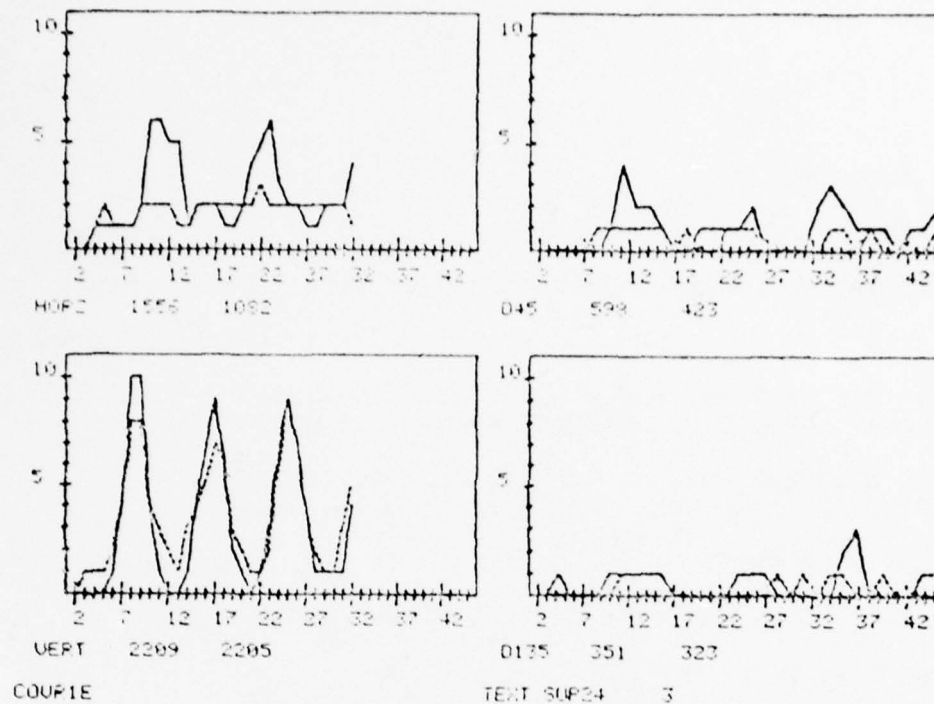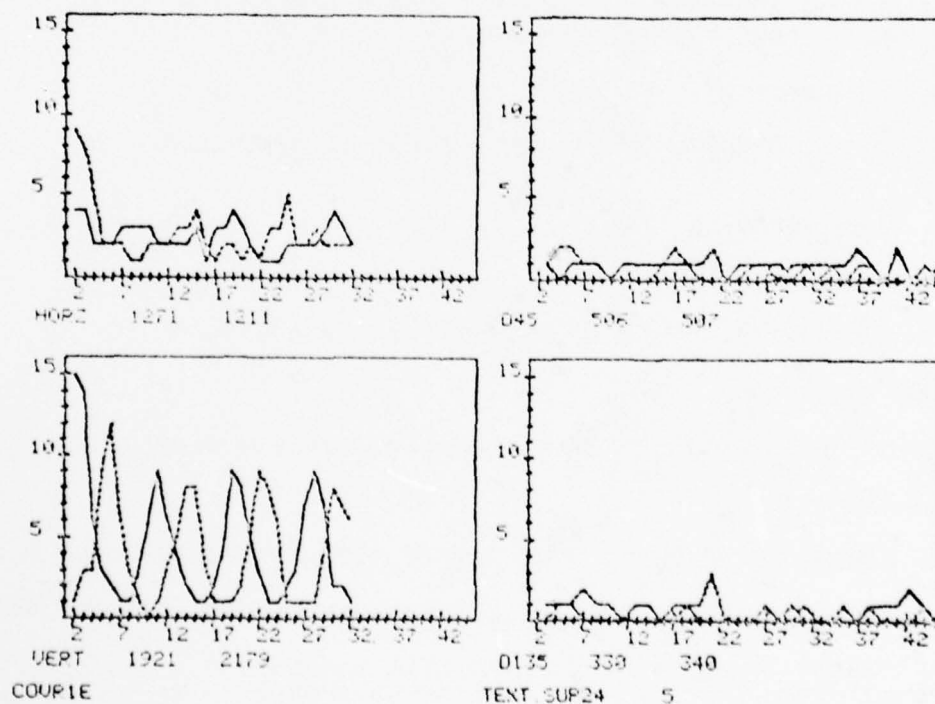
VERTICAL SCAN DIRECTION

   THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 8.000000

   THERE IS STRONG EVIDENCE OF ELEMENT SIZE 6.000000
   WITH STRONG SUPPORT FOR ELEMENT SPACING 8.000000

135 DEGREE SCAN DIRECTION

   NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

Fig. 3. Automatic Description of Raffia in Fig. 1

43

a) 4 Subwindows for Texture Analysis, Raffia on Top,
Herringbone Material on Bottom

b) Non-maximal Suppressed Edges from a

Fig. 4. MOSAIC1
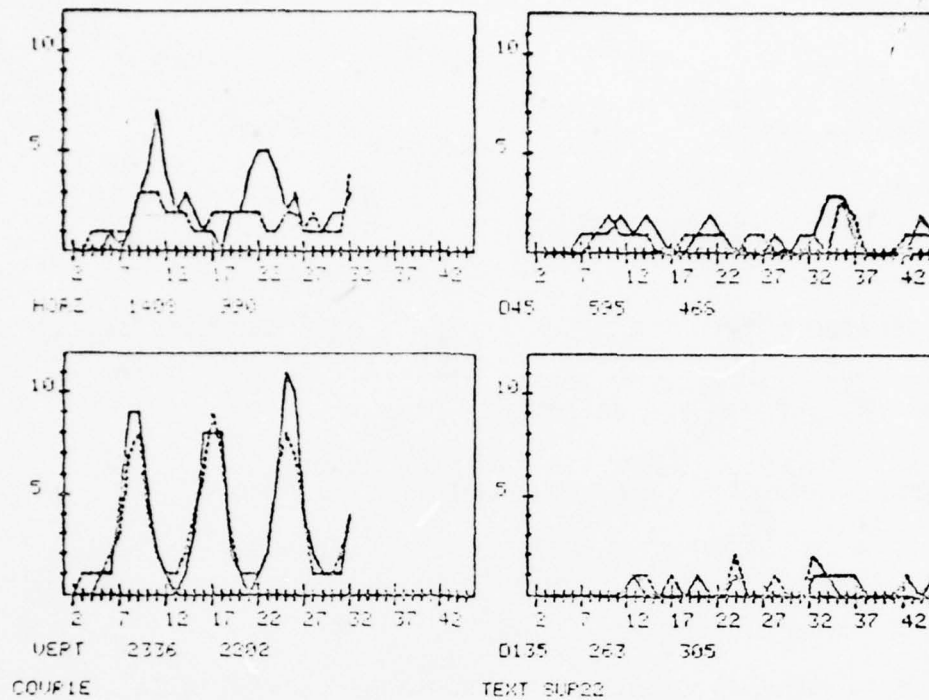
a) RAFFIA.TST22 Element Spacing Graphs
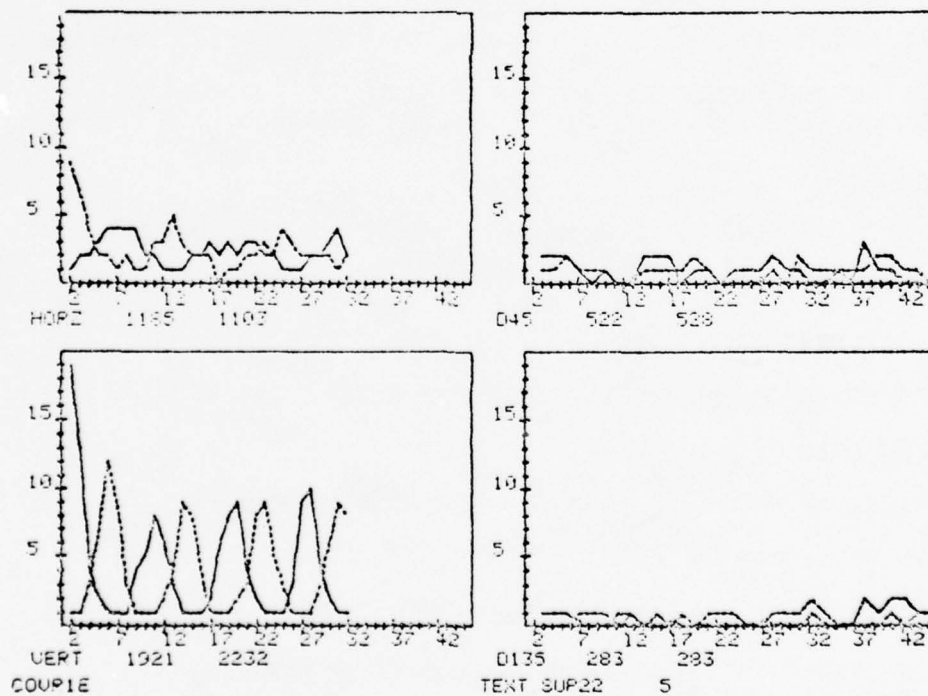
b) RAFFIA.TST 22 Element Size Graphs

Fig. 5.   RAFFIA.TST22 Edge Repetition Results and
Automatic Description

FILENAME = RAFFIA.TST22

DARK OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 11.00000

    THERE IS WEAK EVIDENCE OF ELEMENT SIZE 7.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

    THERE IS WEAK EVIDENCE OF ELEMENT SIZE 8.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

    THERE IS WEAK EVIDENCE OF ELEMENT SIZE 9.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 8.000000

    THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 9.000000

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 2.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 8.000000

135 DEGREE SCAN DIRECTION

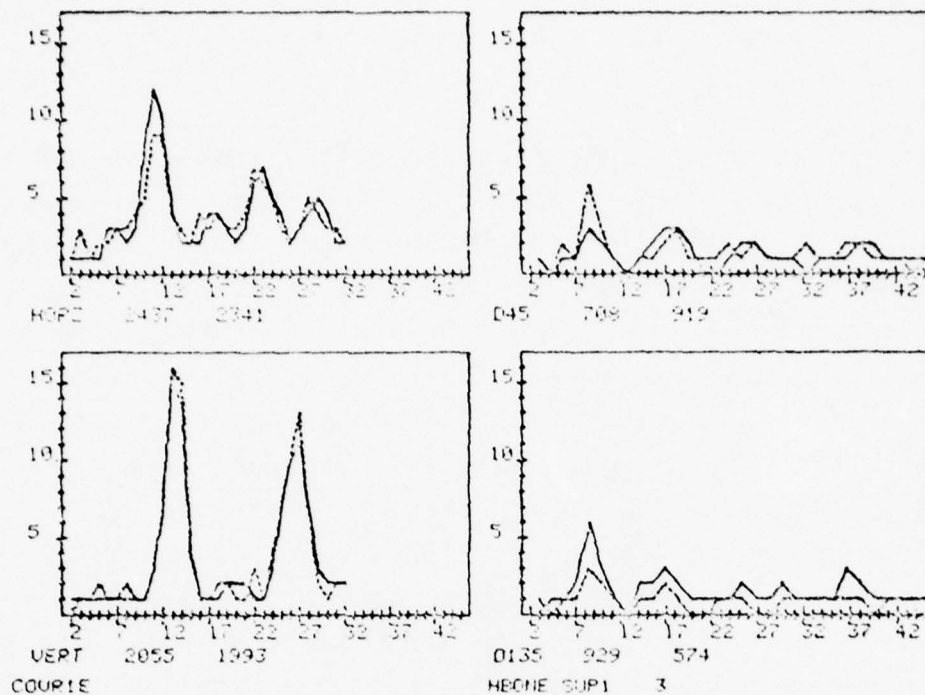   NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

LIGHT OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 2.000000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 9.000000

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 6.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 9.000000

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

c). Automatic Description of RAFFIA.TST22

Fig. 5. CONTINUED

a)   RAFFIA.TST24 Element Spacing Graphs

b)   RAFFIA.TST24 Element Size Graphs

Fig. 6.   RAFFIA.TST24 Edge Repetition Results and
          Automatic Description

47

FILENAME = RAFFIA.TST24

DARK OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 11.00000

    THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 12.00000

    THERE IS WEAK EVIDENCE OF ELEMENT SIZE 7.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 11.000000

    THERE IS WEAK EVIDENCE OF ELEMENT SIZE 9.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 11.000000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 8.000000

    THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 9.000000

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 2.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 8.000000

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

LIGHT OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 2.000000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 8.000000

    THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 9.000000

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 6.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 8.000000

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

c)   Automatic Description of RAFFIA.TST24

Fig. 6. CONTINUED

The most significant results for herringbone material are seen in the horizontal directions. Major element spacing peaks repeat at multiples of 11 and 13 for both light and dark objects in the horizontal and vertical scan directions respectively (see Figs. 7a, 8a). The element size graphs for herringbone material show that there are 2 predominant element size peaks which repeat at intervals of 11 in the horizontal scan direction. These peaks are found roughly at element size values 4 and 8 for dark objects and 3 and 9 for light objects (see Figs. 7b, 8b). Likewise, there are 2 predominant element size peaks which repeat at intervals of 13 in the vertical scan direction. These peaks are found roughly at element size values 3 and 10 for dark objects and 3 and 9 for light objects (see Figs. 7b, 8b). The automatic descriptions of herringbone material given in Figures 7c, 8c indicate that there is evidence that the herringbone material samples are regular with spacing 11 in the horizontal scan direction and 13 in the vertical scan direction; and that the predominant element sizes for dark and light objects in these scan directions are found at approximately element size values 3 and 10.

The mosaic pictured in figure 9a contains 2 sample images of wood in the top row and 1 sample each of water and San Francisco in the bottom row. Figure 9b contains the corresponding edge images for these samples.

The most significant results for wood and water occur in the horizontal scan direction. The most salient characteristic of the horizontal graphs of these samples in the major peak occurring at element size value 2. (see Figs. 10b, 11b, 12b). Each of the automatic descriptions (Figs. 10c, 11c, 12c) note strong evidence for element size 2 for for dark and light object in the horizontal scan direction. An element spacing value of 4 is strongly indicated for both dark and light objects in the horizontal scan direction of the water subwindow; while only 1 of the wood subwindow descriptions indicates evidence of periodicity and this only for light object in the horizontal scan direction.

49

a)   HBONE.TST1 Element Spacing Graphs
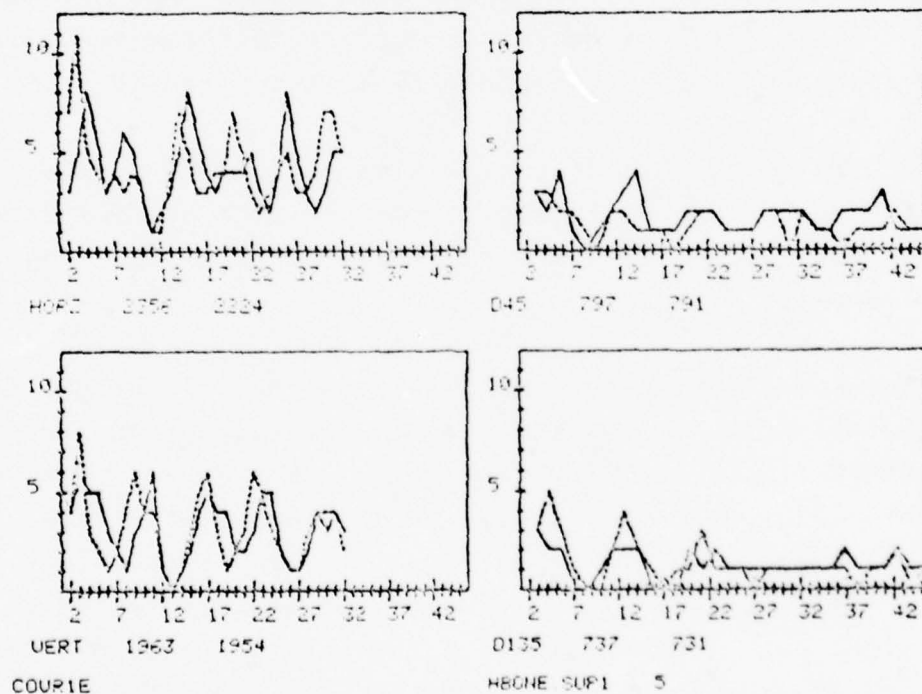
b)   HBONE.TST1 Element Size Graphs

Fig. 7.   HBONE.TST1 Edge Repetition Results and
          Automatic Description

50

FILENAME = HBONE.TST1

DARK OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 11.00000

    THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 4.000000
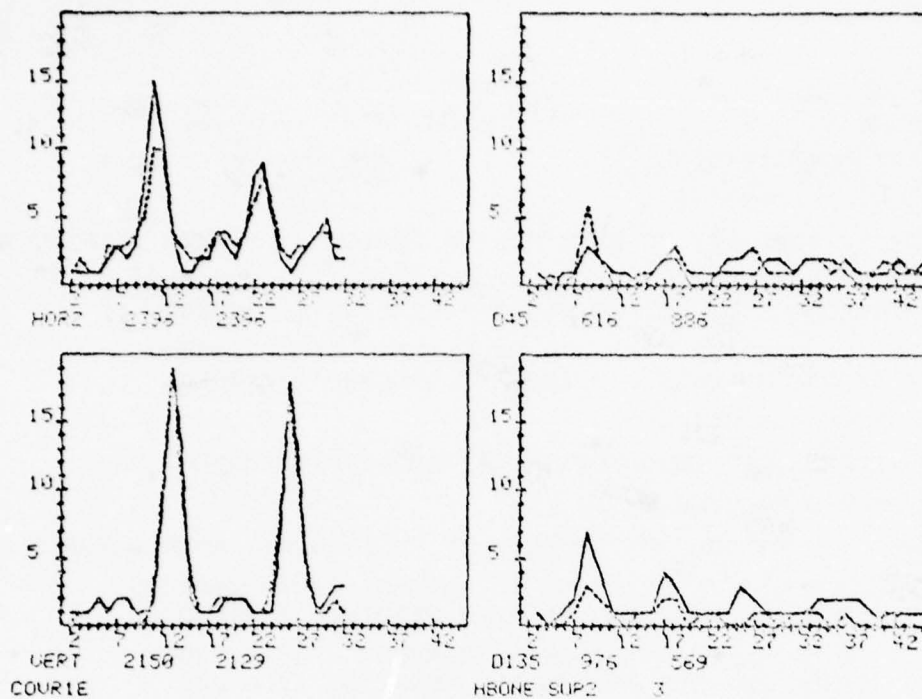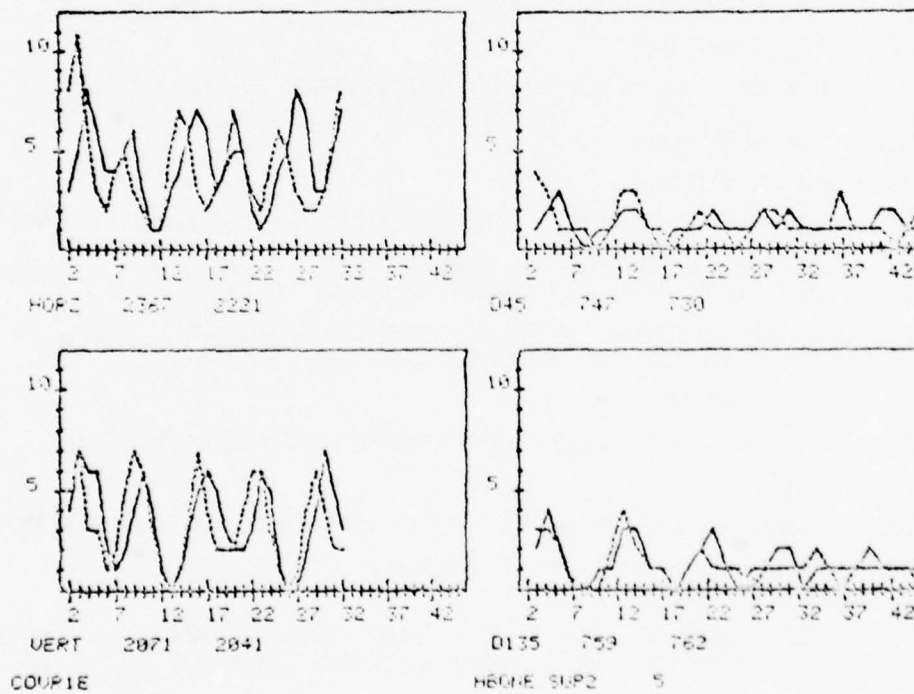   WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

    THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 8.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 13.00000

    THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 3.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000

    THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 4.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000

    THERE IS WEAK EVIDENCE OF ELEMENT SIZE 5.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000

    THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 11.00000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

LIGHT OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 11.00000

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 3.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

    THERE IS WEAK EVIDENCE OF ELEMENT SIZE 9.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 13.00000

    THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 3.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000

    THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 9.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000
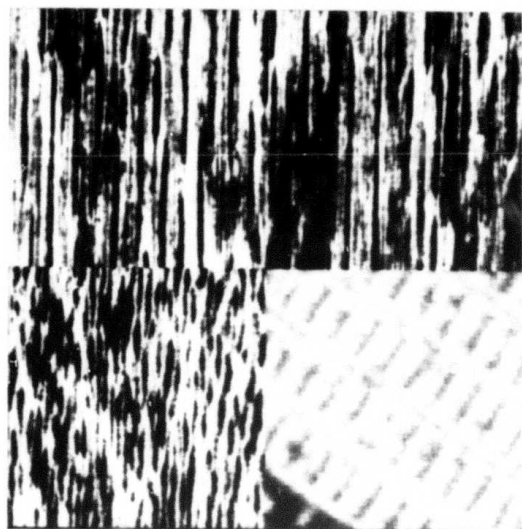
135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY
   WEAK EVIDENCE OF ELEMENT SIZE OF 4.242641

    WEAK EVIDENCE OF ELEMENT SIZE OF 12.72792

                c)   Automatic Description of HBONE.TST1
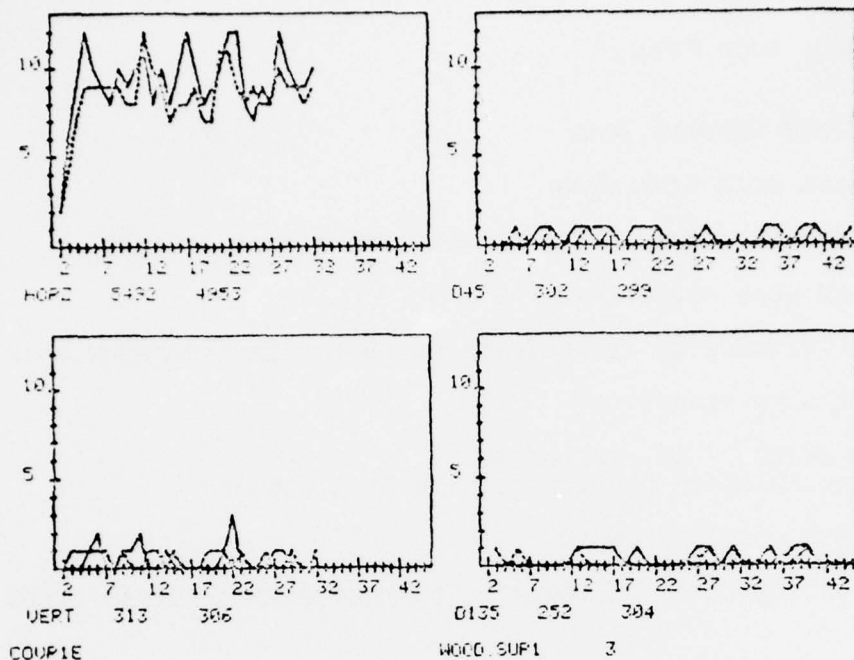
                    Fig. 7. CONTINUED

a)   HBONE.TST2 Element Spacing Graphs

b)   HBONE.TST2 Element Size Graphs

Fig. 8.   HBONE.TST2 Edge Repetition Results and
          Automatic Description

```
FILENAME = HBONE.TST2
```

DARK OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 11.00000

THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 4.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 8.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 9.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

45 DEGREE SCAN DIRECTION

NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION
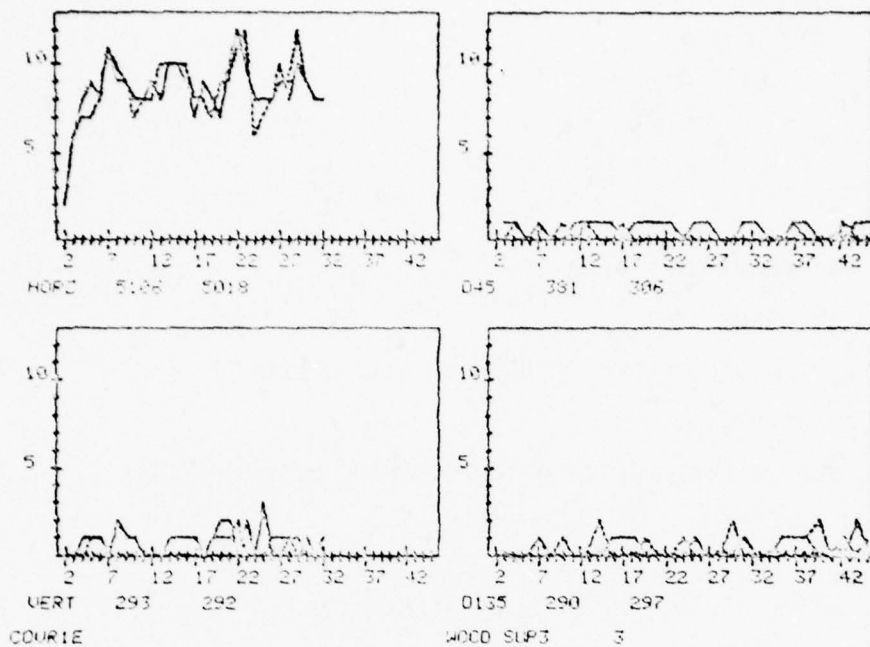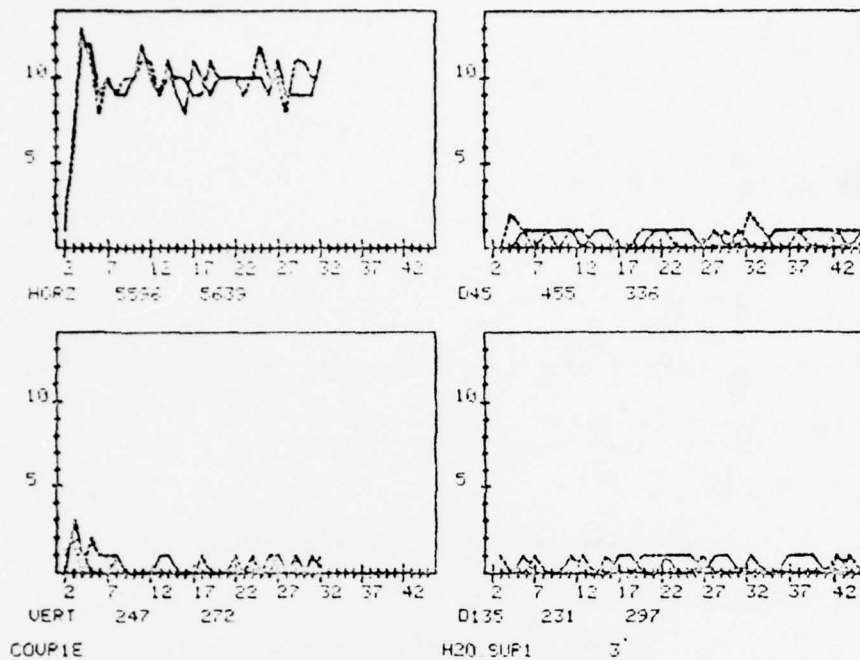
THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 13.00000

THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 3.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000

THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 4.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000

THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 10.00000
WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000

135 DEGREE SCAN DIRECTION

THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 8.485282

THERE IS WEAK EVIDENCE OF ELEMENT SIZE 4.242641
WITH STRONG SUPPORT FOR ELEMENT SPACING 8.485282

LIGHT OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 11.00000

THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 12.00000

THERE IS STRONG EVIDENCE OF ELEMENT SIZE 3.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 8.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 11.00000

45 DEGREE SCAN DIRECTION

NO EVIDENCE OF PERIODICITY
WEAK EVIDENCE OF ELEMENT SIZE OF 2.828427

VERTICAL SCAN DIRECTION

THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 13.00000

THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 3.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000

THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 9.000000
WITH STRONG SUPPORT FOR ELEMENT SPACING 13.00000

135 DEGREE SCAN DIRECTION

NO EVIDENCE OF PERIODICITY
WEAK EVIDENCE OF ELEMENT SIZE OF 4.242641

WEAK EVIDENCE OF ELEMENT SIZE OF 12.72792

c)   Automatic Description of HBONE.TST2

Fig. 8. CONTINUED

a) 4 Subwindows for Texture Analysis, Wood on Top,
Water and San Francisco on Bottom
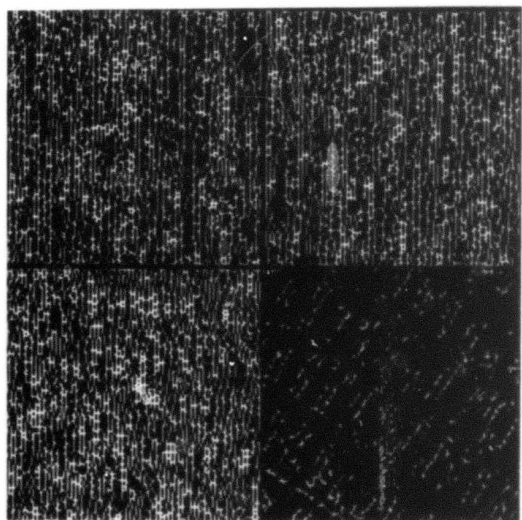
b) Non-maximal Suppressed Edges from a

Fig. 9. MOSAIC2

54

a)   WOOD.TST1 Element Spacing Graphs
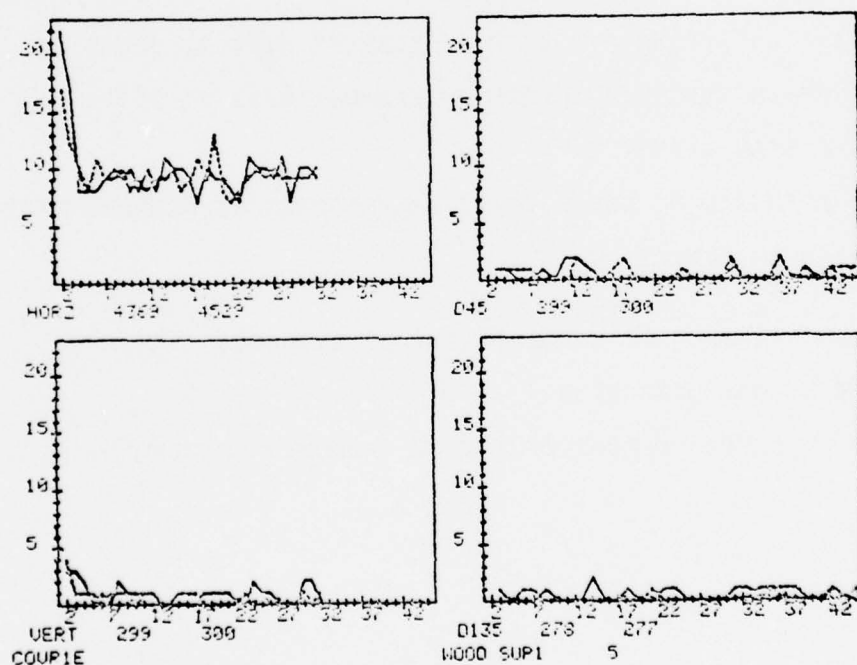
b)   WOOD.TST1 Element Size Graphs

Fig. 10.   WOOD.TST1 Edge Repetition Results and
           Automatic Description

55

FILENAME = WOOD.TST1

DARK OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 2.000000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

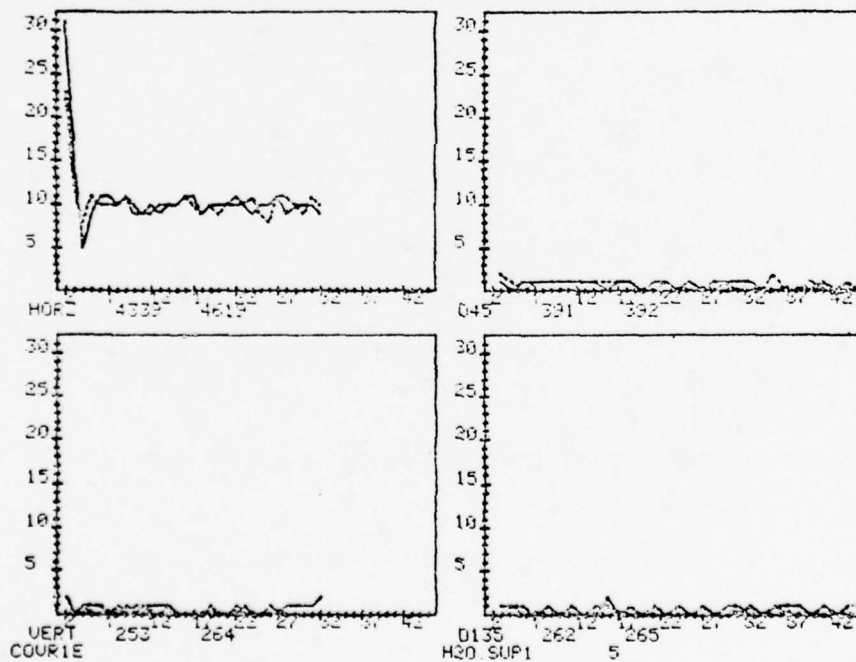   NO EVIDENCE OF PERIODICITY
    WEAK EVIDENCE OF ELEMENT SIZE OF 3.000000

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

LIGHT OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 2.000000

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 20.00000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY
    WEAK EVIDENCE OF ELEMENT SIZE OF 2.000000

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

c)   Automatic Description of WOOD.TST1

Fig. 10. CONTINUED

a) WOOD.TST3 Element Spacing Graphs
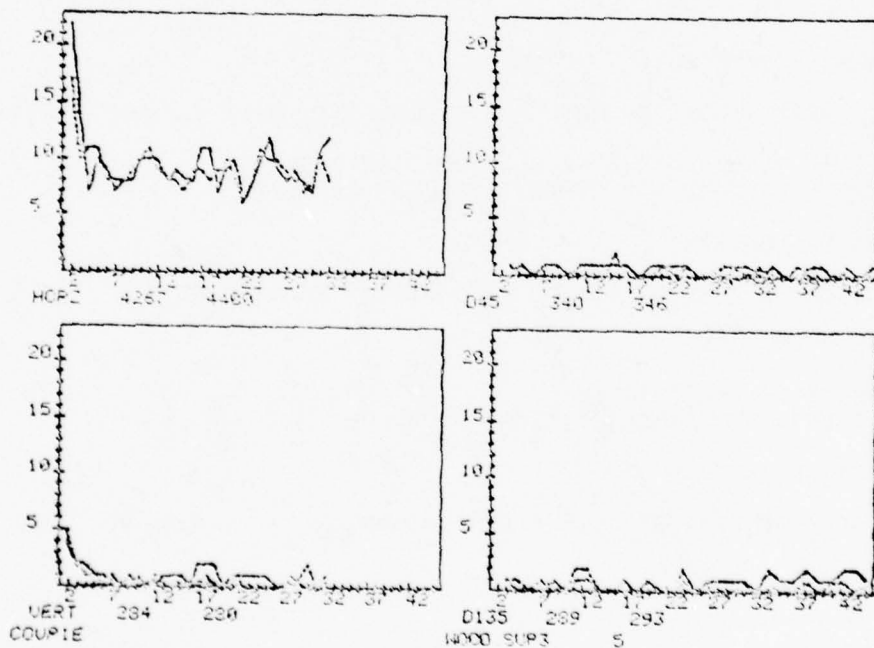
b) WOOD.TST3 Element Size Graphs

Fig. 11.   WOOD.TST3 Edge Repetition Results and
            Automatic Description

FILENAME = WOOD.TST3

DARK OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 2.000000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY
    WEAK EVIDENCE OF ELEMENT SIZE OF 2.000000

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

LIGHT OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 7.000000

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 2.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 7.000000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY
    WEAK EVIDENCE OF ELEMENT SIZE OF 2.000000

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

c) Automatic Description of WOOD.TST3

Fig. 11. CONTINUED

a)   H20.TST1 Element Spacing Graphs

b)   H20.TST1 Element Size Graphs

Fig. 12.   H20.TST1 Edge Repetition Results and
            Automatic Description

```
FILENAME = H20.TST1


DARK OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 4.000000

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 2.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 4.000000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE


LIGHT OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS VERY STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SAPCING 4.000000

    THERE IS STRONG EVIDENCE OF ELEMENT SIZE 2.000000
    WITH STRONG SUPPORT FOR ELEMENT SPACING 4.000000

45 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

VERTICAL SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE
```

c) Automatic Description of H20.TST1

Fig. 12. CONTINUED

As one might expect from looking at the San Francisco subwindow (Fig. 9a) the most significant graphic results occur in the 45 degree scan direction. Element spacing peaks repeat at multiples of ≈15.5 for light and dark objects in this scan direction (Fig. 13a). Support for this element spacing value is given by the element size graphs for the 45 degree scan direction given in Fig. 13b). Moderate peaks at ≈4 for dark objects and ≈10 for light objects repeat at intervals of roughly 15.5.

The automatic description generated for the San Francisco subwindow (Fig. 13c) notes evidence of periodicity with element spacing value ≈15.5. It also notes that dark objects of size ≈4 and light objects of ≈9.8 repeat with element spacing 15.5.

Conclusions

The results presented here are preliminary, but hopefully indicate a step towards useful structural descriptions. In future work, we hope to improve the description process and evaluate the resulting descriptions using more test samples.

References

1. K. Price, R. Nevatia and F. Vilnrotter, "Describing Natural Textures," USCIPI Semiannual Tech Report 860, March 1979, pp. 29-54.

2. R. Nevatia, K. Price and F. Vilnrotter, "Describing Natural Textures," in Proc. IJCAI-79, Tokyo, Japan, Aug. 1979.
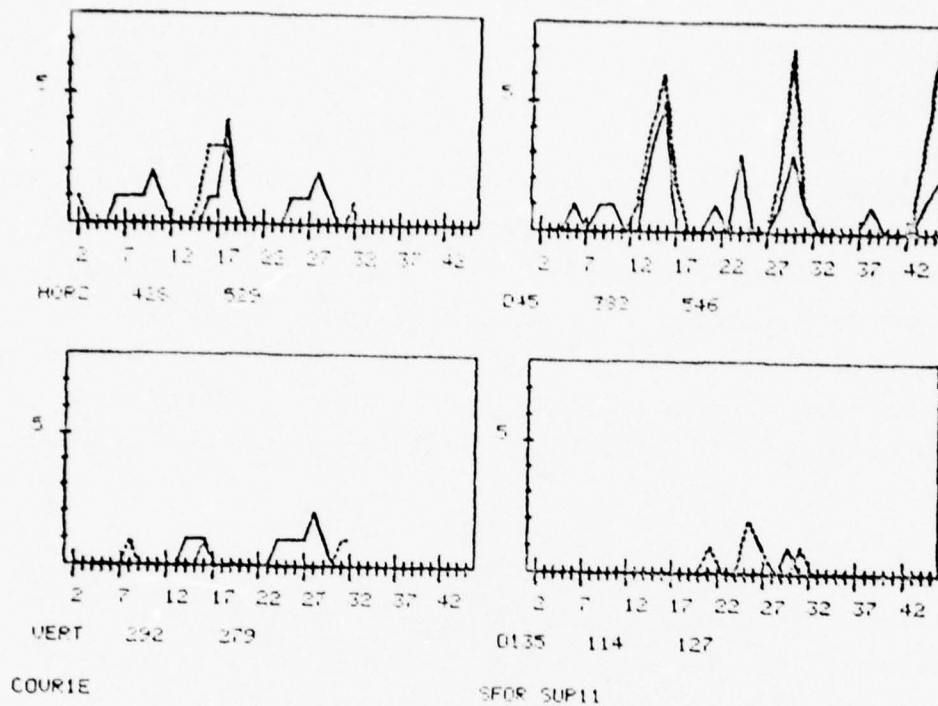
a)   SFOR.TST Element Spacing Graphs
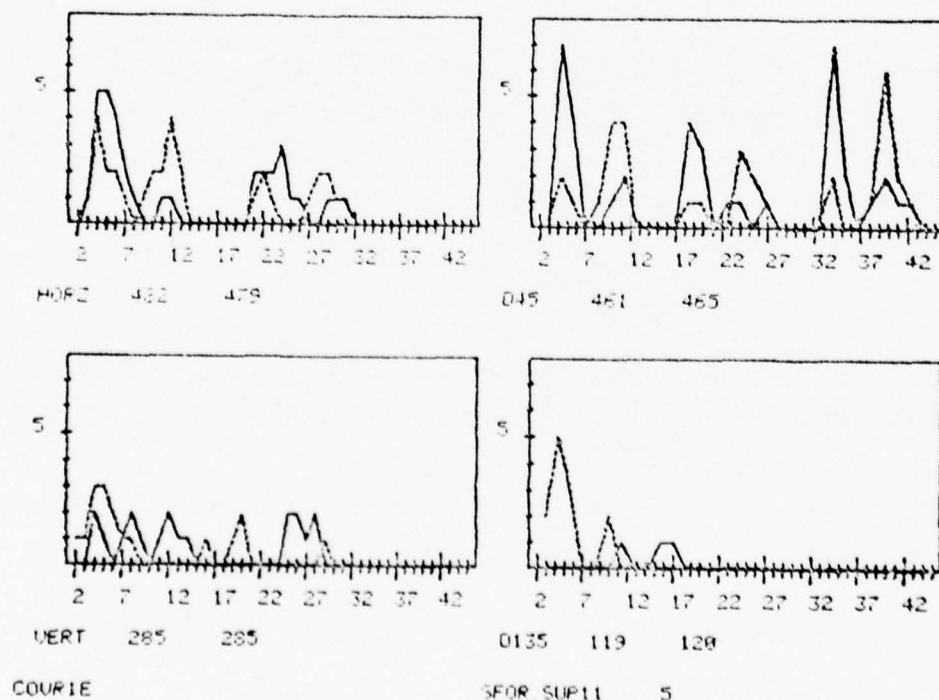
b)   SFOR.TST Element Size Graphs

Fig. 13.   SFOR.TST (San Francisco) Edge Repetition Results
            and Automatic Description

62

FILENAME = SFOR.TST

DARK OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY
    WEAK EVIDENCE OF ELEMENT SIZE OF 4.000000

45 DEGREE SCAN DIRECTION

    THERE IS MODERATE EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 15.55635

    THERE IS MODERATE EVIDENCE OF ELEMENT SIZE 4.242641
    WITH STRONG SUPPORT FOR ELEMENT SPACING 15.55635

VERTICAL SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

LIGHT OBJECT DESCRIPTIONS

HORIZONTAL SCAN DIRECTION

    THERE IS WEAK EVIDENCE OF ELEMENT SIZE 4.000000

    THERE IS WEAL EVIDENCE OF ELEMENT SIZE 12.00000

45 DEGREE SCAN DIRECTION

    THERE IS STRONG EVIDENCE OF PERIODICITY WITH ELEMENT SPACING 15.55635

    THERE IS WEAK EVIDENCE OF ELEMENT SIZE 4.242641
    WITH STRONG SUPPORT FOR ELEMENT SPACING 15.55635

    THERE IS WEAK EVIDENCE OF ELEMENT SIZE 9.899495
    WITH STRONG SUPPORT FOR ELEMENT SPACING 15.55635

VERTICAL SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY OR PREDOMINANT ELEMENT SIZE

135 DEGREE SCAN DIRECTION

    NO EVIDENCE OF PERIODICITY
    MODERATE EVIDENCE OF ELEMENT SIZE OF 4.242641

            c)   Automatic Description of SFOR.TST

               Fig. 13. CONTINUED

2.4 Probability Density Function of the Singular Values of a Random Texture Field

   Behnam Ashjari

   Singular values of a sample matrix or picture can be considered as descriptor or features of the matrix or picture elements and their inter-relationships. Although singular value decomposition, in the recent years, has gained applications and importance, there has not been any statistical treatment of the subject. What follows is the results of a systematic treatment of this subject. For details and derivations, the reader can refer to [1].

Singular Value Decomposition

   A general kxn matrix $\underline{F}$ can be decomposed into three matrices as follows:

$$\underline{F} = \underline{U}\ \underline{S}\ \underline{V}^T \tag{1}$$

where, $\underline{U}$ is a kxk orthogonal matrix such that

$$\underline{U}\ \underline{U}^T = I_k \tag{2}$$

and $\underline{V}$ is a nxk orthogonal matrix such that

$$\underline{V}\ \underline{V}^T = I_n \tag{3}$$

and, $\underline{S}$ is a diagonal matrix whose elements are real, and positive and are called singular values of $\underline{F}$. If the elements of $\underline{S}$ are arranged according to a descending order, then the decomposition (1) will be unique. It can be shown that

64

$$\underline{F} \ \underline{F}^T = \underline{U} \ \underline{\Lambda} \ \underline{U}^T \tag{4}$$

and,

$$\underline{F}^T \ \underline{F} = \underline{V} \ \underline{\Lambda} \ \underline{V}^T \tag{5}$$

where,

$$\underline{\Lambda} = \underline{S}^2 \tag{6}$$

Zonal Polynomials of the Group Representation Theory

The concept of zonal harmonics of a positive definite symmetric matrix has been introduced in [2]. A detailed study of this concept has been presented in [3]. Consider a KxK positive definite, symmetric matrix $\underline{A}$. Let V_p be the vector space of homogeneous polynomials of degree P in the distinct elements of $\underline{A}$. It can be proved that V_p decomposes into a direct sum of irreducible invariant subspaces $V_{\rlap{/}p}$. $\rlap{/}p$ is a partition of P into not more than k parts such that $p_1 + p_2 + \ldots + p_k = p$ and $p_1 \geq p_2 \geq \ldots \geq p_k$. For every value of p, there are a number of possible partitions shown as $\{\rlap{/}p\}$. The polynomial $(tr\underline{A})^P \epsilon V_p$ has a unique decomposition into polynomials $C_{\rlap{/}p}(\underline{A}) \epsilon V_{\rlap{/}p}$ represented as in (7):

$$(tr\underline{A})^P = \sum_{\rlap{/}p} C_{\rlap{/}p}(\underline{A}) \tag{7}$$

where, $C_{\rlap{/}p}(\underline{A})$ is the zonal polynomial of matrix $\underline{A}$ with respect to the partition $\rlap{/}p$ of P and $\sum_{\rlap{/}p}$ means summation over $\{\rlap{/}p\}$. Zonal polynomials can be obtained analytically and in the following derivations, they can be evaluated computationally up to a desired degree of accuracy. For a more detailed introduction to the zonal representation of a

symmetric space, the reader is referred to [1].

Probability Density Function of F

Let $\underline{F}$ be a kxn $(k \leq n)$ matrix whose elements are normally distributed with correlation in column and row directions. Let $\underline{K}_C$ and $\underline{K}_R$ be covariance matrices for column and row directions. And, let

$$g(\underline{F}) \equiv \text{p.d.f. of } \underline{F} \tag{8}$$

and,

$$dG(\underline{F}) \equiv \text{distribution of } \underline{F} = g(\underline{F})d(\underline{F}) \tag{9}$$

Then, p.d.f. of $\underline{F}$ is obtained as

$$g(\underline{F}) = \frac{1}{(2\pi)^{kn/2}|\underline{K}_C|^{n/2}|\underline{K}_R|^{k/2}} \exp\{-\frac{1}{2}\text{tr}(\underline{K}_C^{-1}\underline{F}\underline{K}_R^{-1}\underline{F}^T)\} \tag{10}$$

Differential Decomposition of F

Using differential operations on decomposition of a matrix [4], and using equation (1), we can obtain

$$dF = \frac{2^k \pi^{(\frac{k^2}{2} + \frac{kn}{2})}}{\Gamma_k(\frac{k}{2})\Gamma_k(\frac{n}{2})} (\prod_{i=1}^{K} s_i)^{n-k} \prod_{i<j}^{K} (s_i^2-s_j^2)d(\underline{S})d(\underline{U})d(\underline{V}) \tag{11}$$

where, $\underline{\Gamma}_k(x)$ is K-variate gamma function of x and,

$$\underline{S} = \text{diag}(s_1, s_2, \ldots, s_k) \tag{12}$$

also,

$$d(\underline{S}) = \prod_{i=1}^{k} d(s_i) \tag{13}$$

the differential form $d(\underline{U})$ is the normalized invariant or Haar measure on the orthogonal group $O(k)$ and the differential form $(d\underline{V})$ is the normalized invariant measure on the stiefel manifold of k-frames in space R^n.

Joint Probability Density Function of Singular Values of F

Using relations (7) to (13) and integrating out $\underline{U}$ and $\underline{V}$ over the orthogoal group $O(k)$ and Stiefel manifold of k-frame in R^n w.r.t. the invariant measures $d(\underline{U})$ and $d(\underline{V})$. The joint p.d.f. of the singular values of $\underline{F}$ is obtained as

$$g(s_1, \ldots, s_k) = \frac{2^{k(1-\frac{n}{2})} \pi^{k^2/2}}{\Gamma_k(\frac{k}{2}) \Gamma_k(\frac{n}{2}) |\underline{K}_C|^{n/2} |\underline{K}_R|^{k/2}} \prod_{i<j}^{k} (s_i^2 - s_j^2) (\prod_{i=1}^{k} s_i)^{n-k} \times$$

$$\sum_{P=0}^{\infty} \sum_{\not{k}} \frac{C_{\not{k}}(-\frac{1}{2}\underline{K}_C^{-1}) C_{\not{k}}(\underline{K}_R^{-1}) C_{\not{k}}(\underline{S}^2)}{P! C_{\not{k}}(I_K) C_{\not{k}}(I_n)} \tag{14}$$

67

Joint Probability Density Function of Eigen Values of FF^T

Jacobian of the transformation (6) is as in (15):

$$J_{\underline{\Lambda}}(s_1,\ldots,s_k) = 2^k \prod_{i=1}^{k} s_i = 2^k \det(\underline{S}) \tag{15}$$

using (15) in (14), we obtain the joint p.d.f. of the eigenvalues of FF^T as

$$g(\lambda_1,\lambda_2,\ldots,\lambda_k) = \frac{\pi^{k^2/2}}{2^{kn/2}\Gamma_k(\tfrac{k}{2})\Gamma_k(\tfrac{n}{2})|\underline{K}_C|^{n/2}|\underline{K}_R|^{k/2}}\left(\prod_{i=1}^{k}\lambda_i\right)^{\frac{n-k-1}{2}}\prod_{i<j}^{k}(\lambda_i-\lambda_j)\ \times$$

$$\sum_{P=0}^{\infty}\sum_{p}\frac{C_p(-\tfrac{1}{2}\underline{K}_C^{-1})C_p(\underline{K}_R^{-1})C_p(\Lambda)}{C_p(I_K)C_p(I_n)P!} \tag{16}$$

Probability Density Function of FF^T Matrix

It can be proved that (see [1]); If p.d.f. of F is known, p.d.f. of $\underline{FF}^T$ can be derived from it by using (17):

$$g(\underline{F}\ \underline{F}^T)\,d(\underline{FF}^T) = \frac{\pi^{kn/2}}{\Gamma_k(\tfrac{n}{2})}\iint_{\underline{FF}^T}^{\underline{FF}^T+\Delta(\underline{FF}^T)} g(\underline{F})\,|\underline{FF}^T|^{n-k-1/2}\,d(\underline{FF}^T) \tag{17}$$

Using $g(\underline{F})$ of (10) in (17), we obtain (18) as the p.d.f. of $\underline{FF}^T$:

$$g(\underline{F}\,\underline{F}^T) = \frac{|\underline{F}\,\underline{F}^T|^{n-k-1/2}}{2^{kn/2}\Gamma_k(\frac{n}{2})|\underline{K}_C|^{n/2}|\underline{K}_R|^{k/2}} \sum_{P=0}^{\infty}\sum_{\underline{\rho}} \frac{C_{\underline{\rho}}(\underline{K}_R^{-1})C_{\underline{\rho}}(-\frac{1}{2}\underline{K}_C^{-1}\underline{F}\,\underline{F}^T)}{\underline{P}!\, C_{\underline{\rho}}(\underline{I}_n)} \tag{18}$$

the important observation from (18) is that if, in (18), we let $\underline{K}_R = \underline{I}_n$, we will obtain Wishart density $W(\underline{F}\,\underline{F}^T, k, n, \underline{K}_C^{-1})$. Therefore, the Wishart distribution becomes a special case of (18). The importance of this result is in that $K_R = I_n$ means independence among the columns of $\underline{F}$ and this is the condition on whose basis the Wishart distribution is derived [5]. If we let $K_R = I_n$ and column dimension $k=1$, then $\underline{K}_C$ will be γ_C^2 and (18) will give us the p.d.f. of a chi-squared random variable with n degrees of freedom.

Probability Density Function of the Largest Singular Value of F

Using lemma 3.3 established in [6] and performing the transformation $h_i^2 = \lambda_i/\lambda_1$ for $i=2,\ldots,k$, we can establish the following lemma

$$\int_{1>h_2>\ldots>h_k>0} (\prod_{i=2}^k h_i)^{2t-k} C_{\underline{\rho}}\begin{pmatrix} 1 & & \\ & h_2^2 & \\ & & \ddots \\ & & & h_k^2 \end{pmatrix} \prod_{i=2}^k (1-h_i^2) \prod_{i<j}^k (h_i^2-h_j^2) \prod_{i=j}^k dh_i = \tag{19}$$

$$\frac{kt+P}{2^{k-1}} \frac{\Gamma_k(\frac{k}{2})}{\pi^{k^2/2}} \frac{(t)_{\underline{\rho}}}{(t+\frac{k+1}{2})_{\underline{\rho}}} \frac{\Gamma_k(t)\Gamma_k(\frac{k+1}{2})}{\Gamma_k(t+\frac{k+1}{2})} C_{\underline{\rho}}(\underline{I}_k)$$

where,

$$(a)_{\not{p}} = \prod_{i=1}^{k} \left(a - \frac{1}{2}(i-1)\right)_{P_i} \quad \text{and} \quad \not{p} = (P_1, P_2, \ldots, P_k) \tag{20}$$

using (14) and (19), the p.d.f. of s_1, i.e. the largest singular value of $\underline{F}$ will be obtained as (21):

$$g(s_1) = \frac{1}{2^{\frac{kn}{2}-1}|\underline{K}_C|^{n/2}|\underline{K}_R|^{k/2}} \frac{\Gamma_k\left(\frac{k+1}{2}\right)}{\Gamma_k\left(\frac{n+k+1}{2}\right)} s_1^{kn-1} \sum_{P=0}^{\infty} \frac{\frac{kn}{2}+P}{P!} \left(\frac{s_1^2}{2}\right)^P \sum_{\not{p}} \frac{\left(\frac{n}{2}\right)_{\not{p}}}{\left(\frac{n+k+1}{2}\right)} \times$$

$$\frac{C_{\not{p}}(-\underline{K}_C^{-1})C_{\not{p}}(\underline{K}_R^{-1})}{C_{\not{p}}(\underline{I}_n)} \tag{21}$$

It can be proved [1] that $g(s_1)$ converges and is upper bounded by a well defined function as

$$g(s_1) \leq \frac{1}{2^{\frac{kn}{2}-1}|\underline{K}_C|^{n/2}|\underline{K}_R|^{k/2}} \frac{\Gamma_k\left(\frac{k+1}{2}\right)}{\Gamma_k\left(\frac{n+k+1}{2}\right)} s_1^{kn-1}\left[\frac{kn}{2}e^{\frac{1}{2}s_1^2 \text{tr}(\underline{K}_C^{-1})} + \frac{1}{2}s_1^2 \text{tr}(\underline{K}_C^{-1})e^{\frac{1}{2}s_1^2 \text{tr}(\underline{K}_C^{-1})}\right] \tag{22}$$

It is conjectured that, for known $\underline{K}_R$ and $\underline{K}_C$, expected value of s_1 can be obtained computationally up to a desired degree of accuracy.

p.d.f. of the Sum of Square of Singular Values of F

Using (18) and some careful calculations, we can obtain the p.d.f. of the sum of square of singular values of $\underline{F}$

$$g\left(\sum_{i=1}^{k} s_i^2\right) = \frac{\left(\sum_{i=1}^{k} s_i^2\right)^{\frac{nk-2}{2}}}{2^{kn/2}\Gamma\left(\frac{kn}{2}\right)|K_C|^{n/2}|K_R|^{k/2}} \sum_{P=0}^{\infty} \sum_{p} \frac{C_p(K_C^{-1}\otimes K_R^{-1})\left(-\frac{1}{2}\sum_{i=1}^{k} s_i^2\right)^P}{C_p(I_n)\, P!} \tag{23}$$

Derivation of Moments Related to F and Singular Values of F

The mean value of FF^T is

$$E\{FF^T\} = K_C \operatorname{tr}(K_R) \tag{24}$$

The mean value of the sum of square of singular values is

$$E\left\{\sum_{i=1}^{K} s_i^2\right\} = \operatorname{tr}(K_C \otimes K_R) = \operatorname{tr}(K_C)\operatorname{tr}(K_R) \tag{25}$$

For n=K, moments of products of the singular values can be derived. Using multivariate gamma function and applying the concept on generalized variance introduced in [7, pp. 171], we can obtain the mth moment of the product of the singular values as

$$E\left\{\left(\prod_{i=1}^{K} s_i\right)^m\right\} = 2^{\frac{km}{2}} |K_C|^{\frac{m}{2}} |K_R|^{\frac{m}{2}} \frac{\Gamma_k\left(\frac{k+m}{2}\right)}{\Gamma_k\left(\frac{k}{2}\right)} \tag{26}$$

the second moment or variance of the product of singular values will be

71

$$E\{ (\prod_{i=1}^{k} s_i)^2 \} = |\underline{K}_C||\underline{K}_R|k! \qquad (27)$$

The preceding results have applications in mathematics, control and communication sciences. Their application in image processing and pattern recognition has been explored in [1]. In the derivations, $\underline{F}$ can be a sample window from a Texture field. The p.d.f. of $\underline{F}$ presented in this report results in the p.d.f. of the singular values of $\underline{F}$ which ultimately results in the p.d.f. of the largest singular value. For known $\underline{K}_C$ and $\underline{K}_R$, and from p.d.f. of the largest singular value, it is possible to computationally obtain the expected value of the largest singular value of $\underline{F}$: the best model for $\underline{K}_C$ and $\underline{K}_R$ is a first order Markov covariance matrix which is of toeplitz form. The eigenvalues of a toeplitz form can be derived and as a result the zonal polynomials $C_\kappa(K_C^{-1})$ and $C_\kappa(K_R^{-1})$ can be obtained and used in equation (21). (21) reaches its steady state value rapidly and hence can be programmed efficiently on a computer. The expected value of the largest singular value provides an upper bound to the average variation of the rest of the singular values. The above computations can be used in designing a texture classifier. It can also be used in Bhattacharyya distance analysis of texture fields.

References

1. Behnam Ashjari, Singular Value Decomposition Image Feature Extraction, Ph.D. Dissertation, USCIPI, Image Processing Institute, USC, Los Angeles, California, to be published in approximately December 1979.

2. E. Cartan, "Les Fonctions Zonales d'un espace symetrique irreductible," in Oeuvre Complètes, Partie 1, Vol. 2, 1952, pp. 1065-1070.

3. A.T. James, "Zonal Polynomials of the Real Positive Definite Symmetric Matrices," Ann. Math., Vol. 74, No. 3, November 1961,

pp. 456-469.

4. A.T. James, "Normal Multivariate Analysis and the Orthogonal Group," Ann. Math. Statist., 25, 1954, pp. 40-75.

5. B. Ashjari and W.K. Pratt, "Singular Value Decomposition Image Feature Extraction," Semiannual Technical Report, USCIPI Report 800, Image Processing Institute, USC, Los Angeles, Calif., March 31, 1978, pp. 72-89.

6. T. Sugiyama, "The Distribution of the Largest Latent Root and Corresponding Latent Vector for Principal Component Analysis," Ann. Math. Statistics, Vol. 37, 1966, pp. 995-1001.

7. T.W. Anderson, An Introduction to Multivariate Statistical Analysis, Wiley, N.Y., 1958.

2.5 Models for Texture Analysis and Synthesis

David Donovan Garber

Introduction

Texture is an important characteristic for the analysis of images. Accordingly, the study of this image feature has led to the proposal of models attempting to describe it [1-4]. These models are for the most part ad hoc as have been texture discrimination techniques [5]. Almost all researches have made no more than a slight attempt to apply their models to real natural textures.

Presented in this short paper are the results of independent texture research which is based upon an extension of earlier work at

USC [6] showing application of models to a variety of real natural textures. Many of the basic ideas derived from work in this area are similar to those proposed in earlier texture studies [4,7,8].

The 2-D Binary Markov Model

In the investigation of natural phenomenon once a researcher collects enough data he tries to imagine a process which accounts for the results. The construction and development of a mathematical model is often the best way this can be done. In some cases the model may be extraordinarily complex, in others, exceedingly simple. In most cases model acceptance cannot be based on "truth" as the true generating phenomenon is too complex or just unknown and so it is based upon model usefulness and "how well it works." Such "working" models for texture are presented here because they have the ability to simulate natural textures.

If one can synthesize and simulate natural textures adequately by using some proposed model the criterion of usefulness and workability for that model is met. A researcher may then apply the model to problems of texture identification and discrimination with justification in a non-ad hoc way until a better texture model is developed.

Many early texture studies involved the use of binary textures generated by one-dimensional Markov processes. Such work was used by Purks and Richards [9] and later extended by Garber [6]. In these one dimensional models a large vector of pixels was generated line by line using a set of generation parameters

$$G_{V_{N+1}}(V_1, V_2, \ldots, V_N)$$

where

$$G_{V_{N+1}}(V_1, V_2, \ldots, V_N) = P(V_{N+1}/V_1, V_2, \ldots, V_N) \tag{1}$$

74

and $P(A/B)$ represents the probability of A given B. In the above notation each V_i represents a generated pixel which has value 0 (black) or 1 (white). Each pixel value, then, depends on the N pixels previous to it. A two dimensional texture image was then formed by breaking up the large vector of pixels into shorter strings and stacking them one on top of the other. This procedure for large images nearly insured image row independence (unless N was large) thereby creating only horizontally oriented textures totally unsuitable for simulating natural two-dimensional textures.

By allowing N to increase exceeding the short string line length, two-dimensional (vertical and horizontal) dependence may be induced into the generating process. A pixel value then depends not only on the pixels previous to it on the same line but also on the pixels above it (see Figure 1). In theory, texture dependence could be extended ad infinitum, however practical considerations concerning the actual generation process show us that 2 generation parameters must be accounted for. As a possible solution to the storage problem we can choose to ignore all but N of the previous pixels in our generation process and we can allow the pattern of the N's to become flexible.

Although it was not explicitly stated by Garber [6] the generation parameters of a texture may be estimated for any given set of N V s. These statistics have the property

$$E[\hat{G}_{V_{N+1}}(V_1, V_2, \ldots, V_N)] = G_{V_{N+1}}(V_1, V_2, \ldots, V_N)$$

where

$$\hat{G}_{V_{N+1}}(V_1, V_2, \ldots, V_N) = \hat{P}(V_1, V_2, \ldots, V_N, V_{N+1}) /$$

$$(\hat{P}(V_1, V_2, \ldots, V_N, 0) + \hat{P}(V_1, V_2, \ldots, V_N, 1)) \tag{2}$$

V_1	..	..	..	V_N	V_{N+1}	

One-dimensional Synthesis

V_1	V_2	..	..	..	..	..
..	..	..	..	..	..	..
..	..	..	..	..	..	..
..	..	..	..	..	..	..
..	..	V_N	V_{N+1}			

Two-dimensional Synthesis

Figure 1
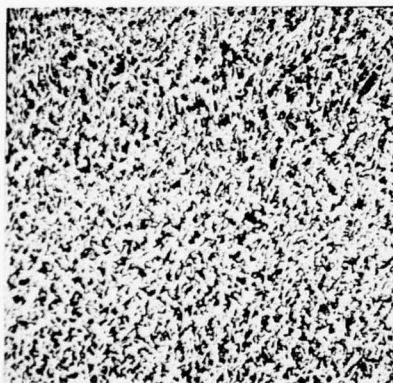
(a)
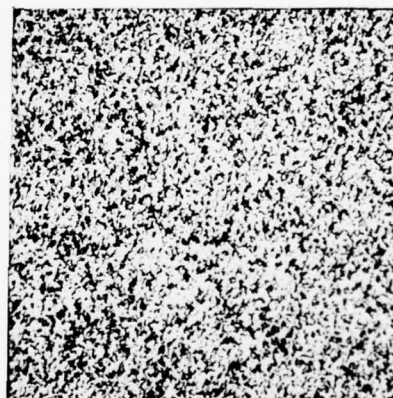
(b)

Figure 2. Grass

(a)

(b)

Figure 3. Bark

76
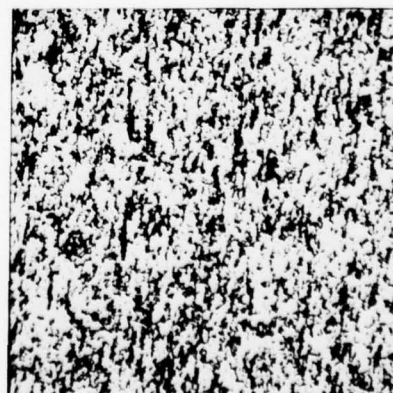
Ignoring boundary conditions, the best linear unbiased estimates ($\hat{P}$'s) of the P's are naturally defined by

$$P(V_1, V_2, \ldots, V_N) = \frac{1}{M} \sum_{j=0}^{M} \prod_{K=1}^{N} \delta(I(K+j) - V_K) \qquad (3)$$

where $I(i)$ represents the ith element of the texture string from which the parameters are to be estimated.

By using this method, generation parameters were estimated from "parent" binary textures and used to synthesize imitation binary textures. (The V_i's were chosen to minimize the sum of squares error assuming a linear model (see next section).) These $(512)^2$ parent textures are illustrated in Figures 2-11(a). Synthesized binary $(512)^2$ simulations are shown in Figures 2-11(b).

As the estimated texture generation parameters are approximated using statistics gathered from the full parent texture, non-homogenity in the parent texture will cause an "average" texture to be synthesized. Because of its non-homogeneous nature (specifically the directionality of the stalks in different parts of the image) and exhibition of detail (specifically individual non-conforming single stalks) the simulation of straw will not be perfect. A similar observation may be made with respect to the parent textures of grass and water but in these textures the non-homogenity is not so pronounced. As we are attempting to synthesize textures and not merely "image code" the parent textures, details and non-homogeneities will be lost in the synthesis process.

Continuous Tone Linear Texture Model

In many problems a grey-scale image is first converted into a binary picture by quantization. As was seen in Figures 2-11, much of the original texture information is retained by such a quantization.
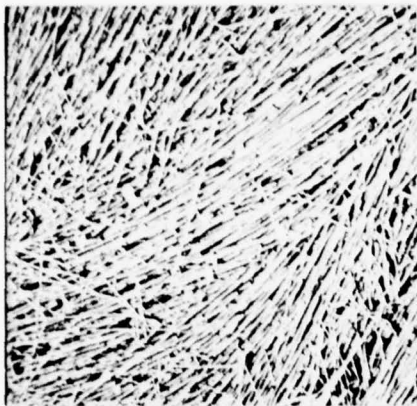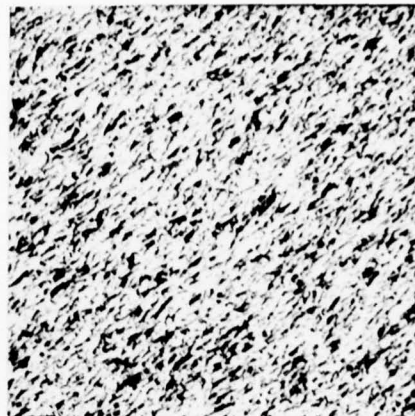
(a)   (b)

Figure 4.   Straw
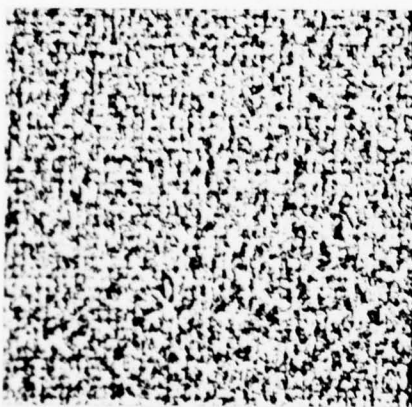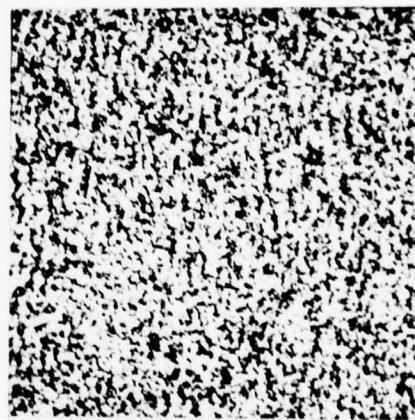
(a)   (b)

Figure 5.   Wool
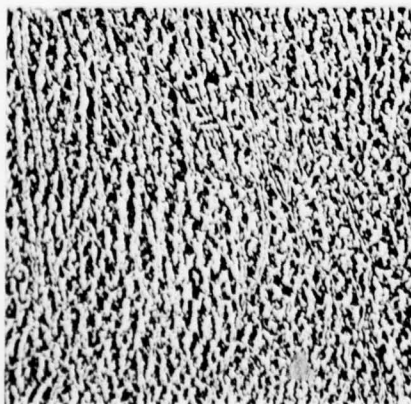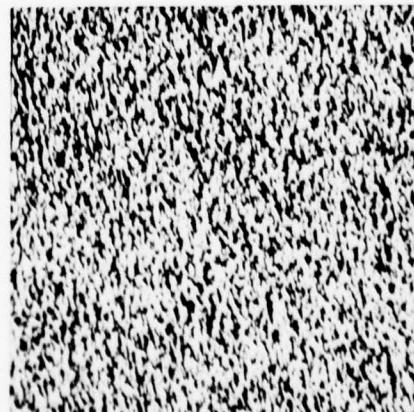
(a)   (b)

Figure 6.   Leather
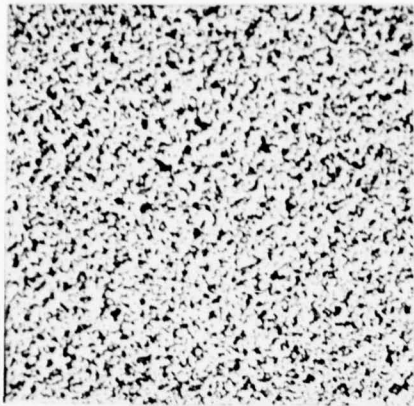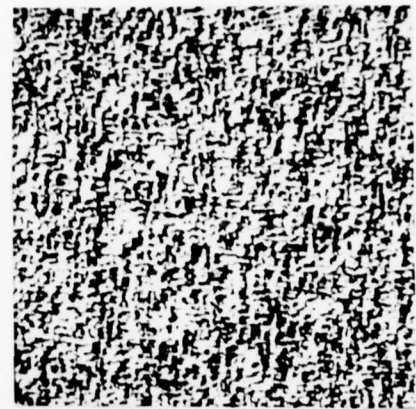
(a)
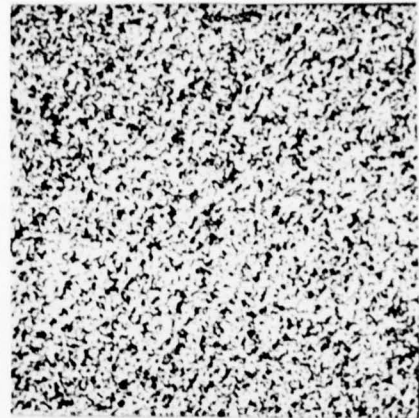
(b)

Figure 7. Sand
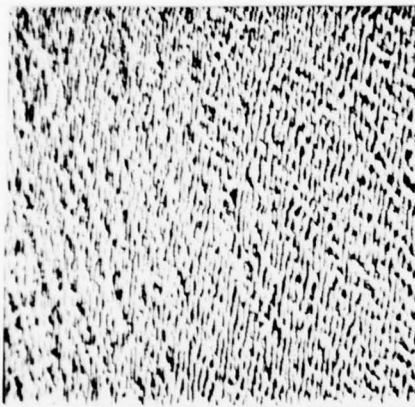
(a)
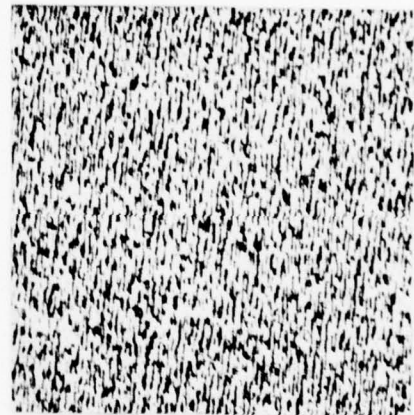
(b)

Figure 8. Water
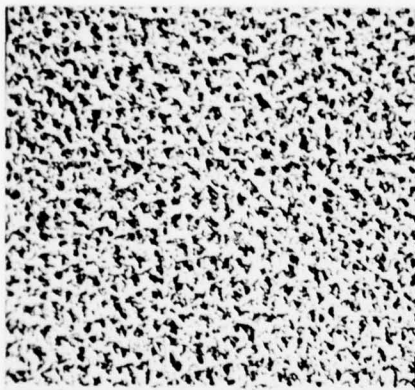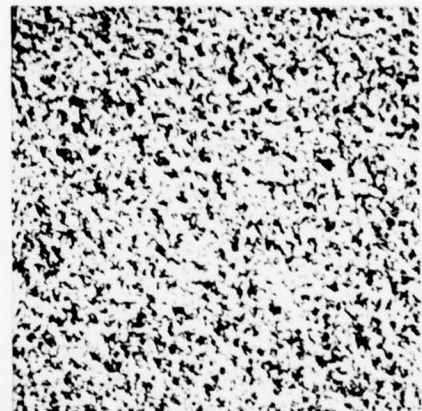

(a)

(b)

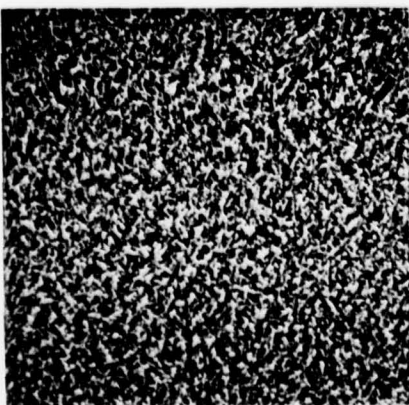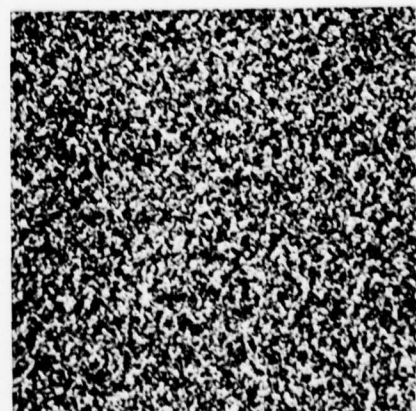Figure 9. Wood

79
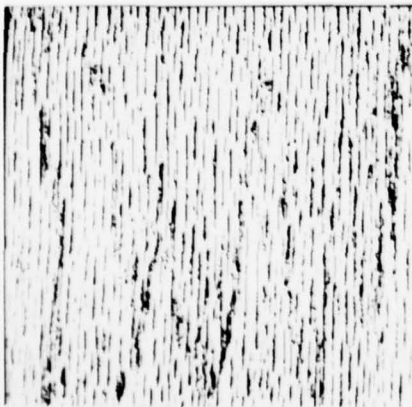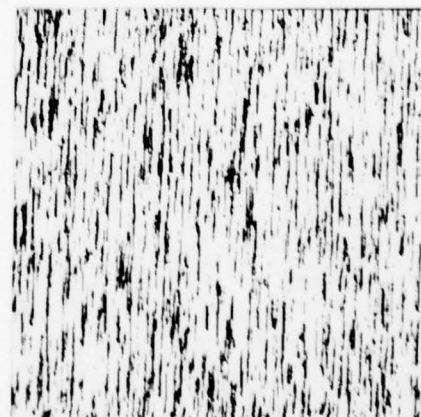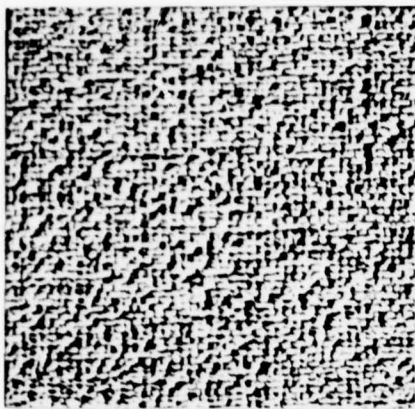
(a)

(b)

Figure 10. Raffia

(a)

(b)

Figure 11. Pigskin

(a)

(b)

Figure 12. Grass

80

Still a model for continuous tone textures is desirable.

As in the binary case, the texture model is closely related to the generation process which successfully simulates it. In the binary case we chose a Markov-N model. A similar model can be chosen for a g-grey level image however the retention and estimation of a full set of generation parameters in this case is impossible as it requires g^N locations (for just 16 levels with N=8 this is over 4×10^9 parameters). After much thought a simple linear model was proposed. This model has also been studied by Tou and Chang, McCormick and Jayaramurthy, and Deguchi and Morishita but their results did not definitively indicate model generality [5]. The model may be expressed as

$$Y = X\beta + \epsilon \tag{4}$$

where

$$Y = V_{N+1} \qquad X = \begin{bmatrix} V_1 \\ V_2 \\ . \\ . \\ . \\ V_N \\ 1 \end{bmatrix}$$

and B is an $(N+1) \times 1$ vector of unobservable parameters, ϵ is an $(N+1) \times 1$ unobservable random vector such that $E[\epsilon]=0$, $0 \leq V_i < g$. The most common estimator of β is $(X'X)^{-1}X'Y$, the least squares estimator. So given a parent texture an estimation of B, $\hat{B}$, may be obtained yielding the generation model

$$\hat{Y} = X\hat{\beta} \tag{5}$$

and noise can be added to our generation process at will. After carefully choosing X and estimating $\hat{B}$ for each 512^2 parent texture shown in Figure 12-20(a) 512^2 synthesized Monte Carlo simulations were generated shown in figures 12-20(b).
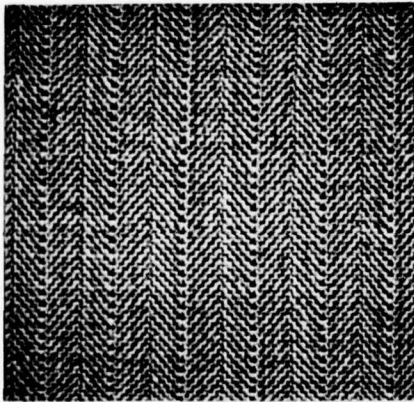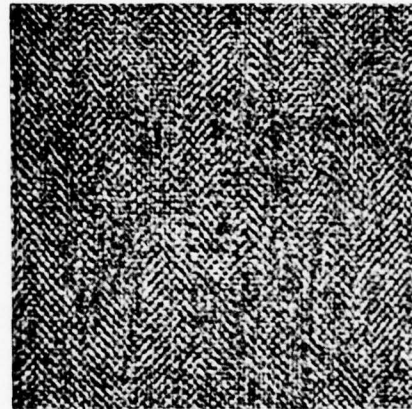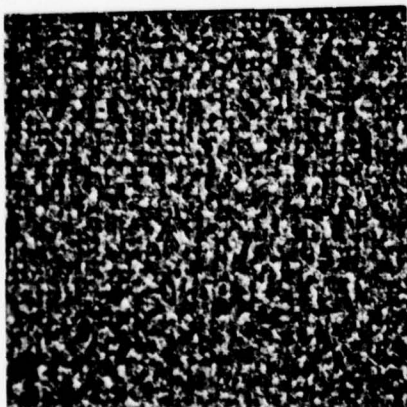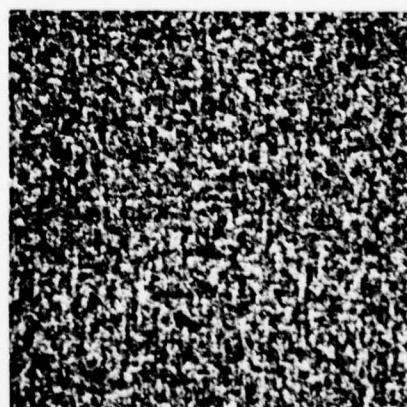
81
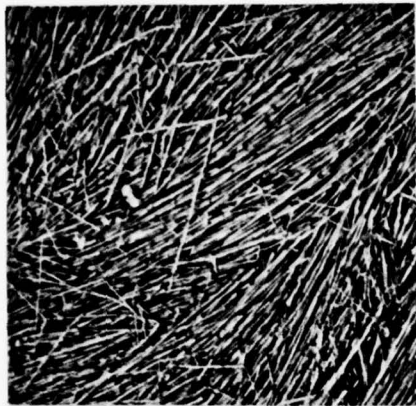
(a)

(b)

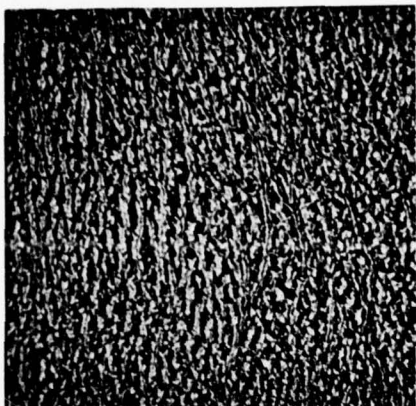Figure 13. Straw

(a)

(b)

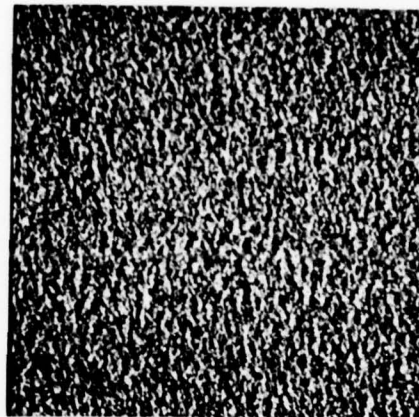Figure 14. Cloth

(a)

(b)

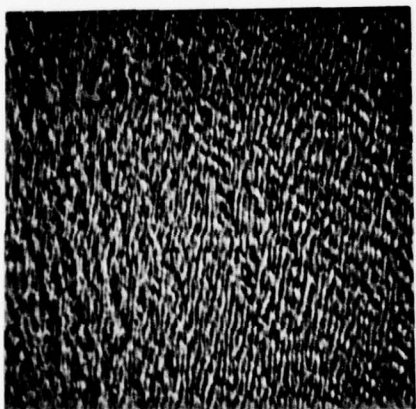Figure 15. Wool

82

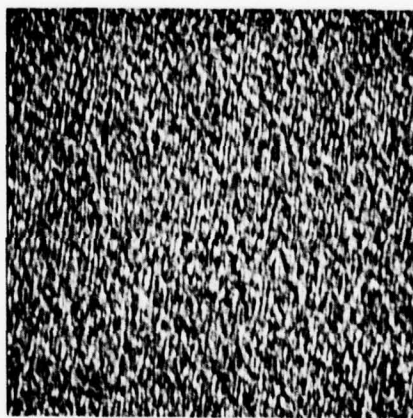(a)   (b)

Figure 16.   Leather
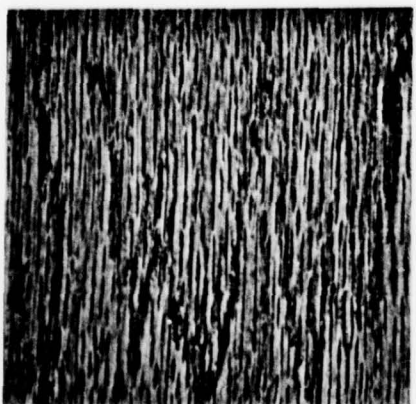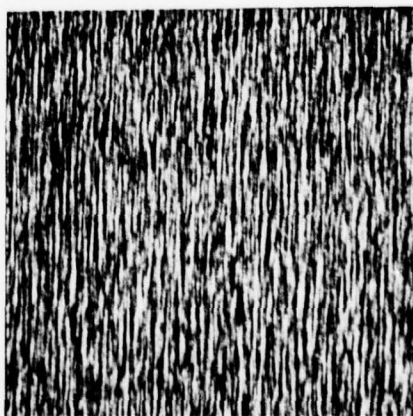

(a)   (b)

Figure 17.   Water

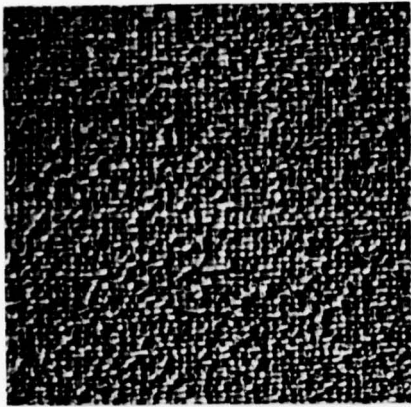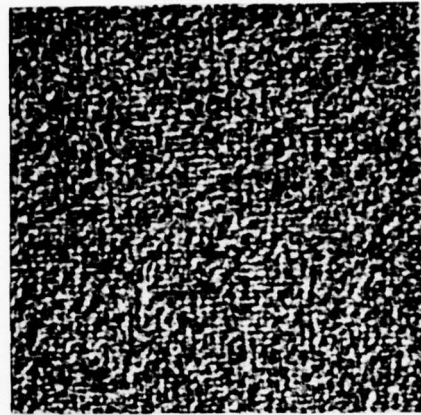(a)   (b)
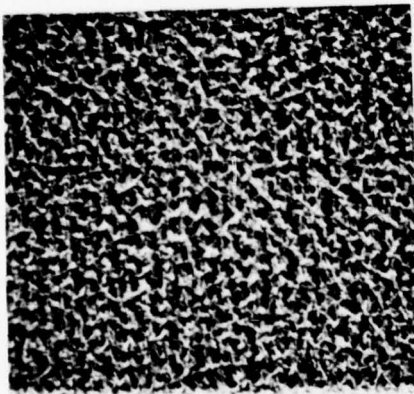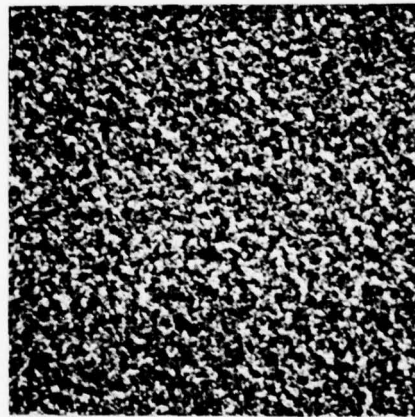
Figure 18.   Wood

(a)   (b)

Figure 19.   Raffia

(a)   (b)

Figure 20.   Pigskin

Conclusions

By expanding the above models and experimenting with the V as well as the noise parameters, improvements in the appearance of the synthesized textures could be obtained at additional compute cost. It would probably be quite a challenge to develop a segmentor powerful enough to separate the simulations from their parents. This is illustrated by passing a 1/4"-square (55 pixels) window over the texture pairs and noting the similarities.

Based upon the results of this work, it is suggested that the auto regression and Markov models should be exercised more closely with respect to their application in texture analysis, identification and synthesis in future study. Such work and more details of the work sketched in this paper will appear in later publications.

References

[1] A. Rosenfeld and L.S. Davis, "Image Segmentation and Image Models," Proc. IEEE, Vol. 67, pp. 764-772, May 1979.

[2] S.W. Zucker, "Toward a Model of Textures," Computer Graph. Image Processing, Vol. 5, No. 2, pp. 190-202, 1976.

[3] M. Hassner and J. Sklansky, "Markov Random Fields as Models of Digitized Image Textures," Proc. Conf. on Pattern RECOGNITION and Image Processing, Chicago, May 1978, pp. 346-351.

[4] B.H. McCormick and S.M. Jayaramurthy, "Time Series Model for Texture Synthesis," Int. J. COMPUT. Inform. Sci., Vol. 3, No. 4, pp. 329-343, Dec. 1974.

[5] R.M. Haralick, "Statistical and Structural Approaches to Texture," Proc. IEEE, Vol. 67, pp. 786-804, May 1979.

[6] D.D. Garber, "One Dimensional Texture Pattern Generation and Discrimination," Technical Report USCIPI 840, 1978.

[7] K. Deguchi and I. Morishita, "Texture Characterization and Texture-Baed Image Partitioning using Two-Dimensional Linear Estimation Techniques," IEEE Trans. on Computers, Vol. C-27, pp. 739-745, Aug. 1978.

[8] J.T. Tou, D.B. KAO, and Y.S. Chang, "Pictoral Texture Analysis and Synthesis," in Proc. 3rd Int. Jnt. Conf. on Pattern Recognition, Coronado, Calif., Nov. 1976, pp. 590-590p.

[9] S.R. Purks and W. Richards, "Visual Texture Discrimination Using Random-Dot Patterns," Journal Optical Society of America, Vol. 67, pp. 765-771, June 1977.

2.6 Structural Object Recognition in Aerial Images

   K. Ramesh Babu

   In this report we describe the progress in trying to develop algorithms to recognize objects with structural descriptions in aerial images. This is part of a larger effort at U.S.C. to produce an I.U. System [5].   This work is still in developmental stages; we desribe the method in terms of airport recognition. Based on the model of a scene, airports are approximately located; for precise location, we need to perform recognition based on (a bank of) airport models. Previous work [2] establishes a low-level processing technology that we propose to make use of for higher-level processing.   We have already employed the linear features for describing roads [3] and we

86

address ourselves here to the problem of object recognition.

The Recognition Process

The recognition process makes use of _models_ of the objects to be recognised. In order to generate constraints that will aid in recognition, _features_ _from_ _the_ _image_ that can be matched with various model details are produced. Then, a _search_ procedure will find possible instances of the model in the image by trying to correspond each individual model detail with image features; in general, one finds more than one correspondence for the model due to various factors - inadequacy of the model details used, symmetries, and noise in the detected features. Finally, a set of rules will often be necessary to disambiguate these correspondences and pronounce a final match.

The next sections describe the details of modeling, image feature extraction, and matching.

Modeling

Our model employs data of the following nature:

1. Runways and taxiways are specified as vectors in a (rectangular) coordinate system, i.e., these are directed straight lines with specific position and length. (Fig. 1).

2. Parallelism relationships. These are described by specifying the two runways and the distance of separation between them.

87

Origin of the coordinate system
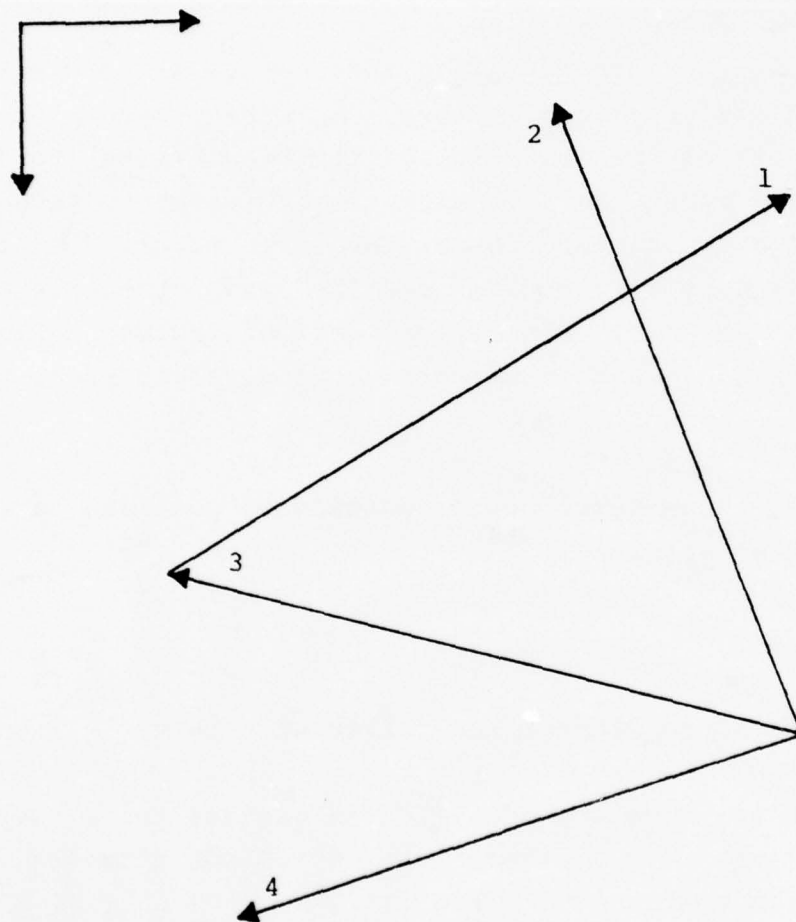

Fig. 1. Example airport model with four runways
         and taxiways

```
type    runway = 1..noofrunways;
        distance = real;
        angle = 0..359;
        parallels = record
                       rl, r2: runway;
                       d: distance;
                    end;
```

3. When runways meet or intersect, they can do so in various ways.
We have described in Figs. 2-4 the more commonly encountered type of
relations (between two runways) -- the X-junction, the Y-junction and
the L-junction. As will be explained later, in the section on
matching, it becomes necessary to distinguish two types of Y-junctions
and four types of X-junctions. When more than two runways participate
in the intersection, the description is a list of pairwise
relationships.   An example will serve to illustrate these points
(Fig. 5).

4. Positional relationships. left of and right of. A runway can be
to the left of another or to the right. Since runways are described
as vectors, these relations can be unambiguously defined.

    A binary intersection relationship is described by the type
intersection:
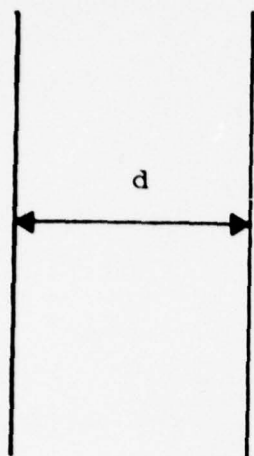
(a) parallel
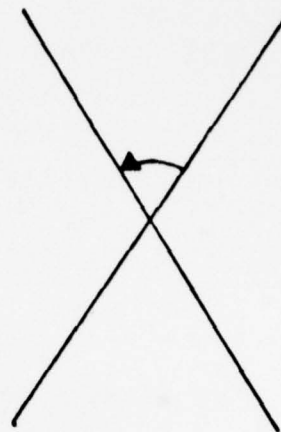
(b) Cross or X-junction
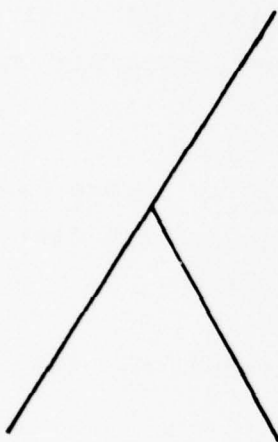
(c) Y-junction
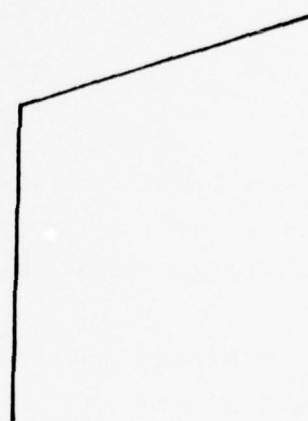
(d) L-junction

Fig. 2. Common relationships among runways of an airport
(directions of runways neglected).
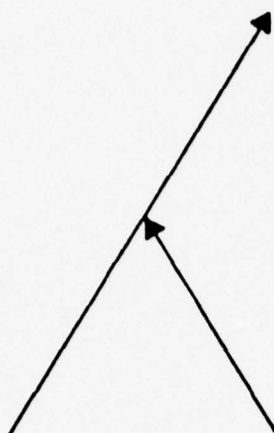
(a)   "bright Y"
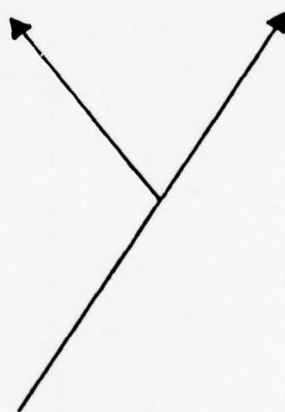
(b)   "dark y"

Fig. 3.   Distinctions between the categories of
          the Y-junction.
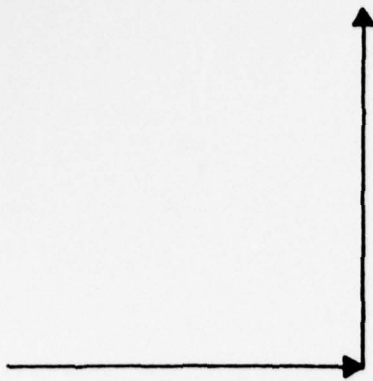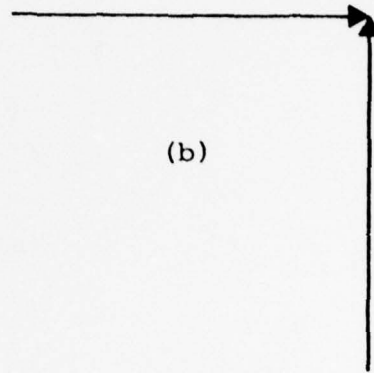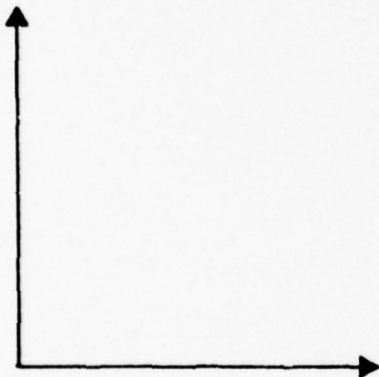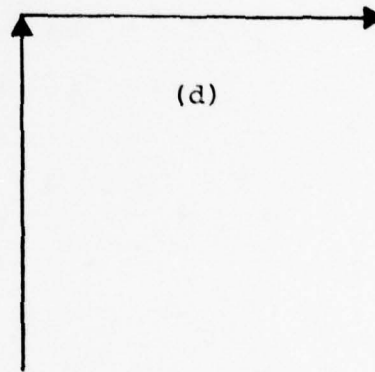
Fig. 4. Distinctions among various categories of the L-junctions.
(a) leftturn (b) converge (c) diverge (d) rightturn.
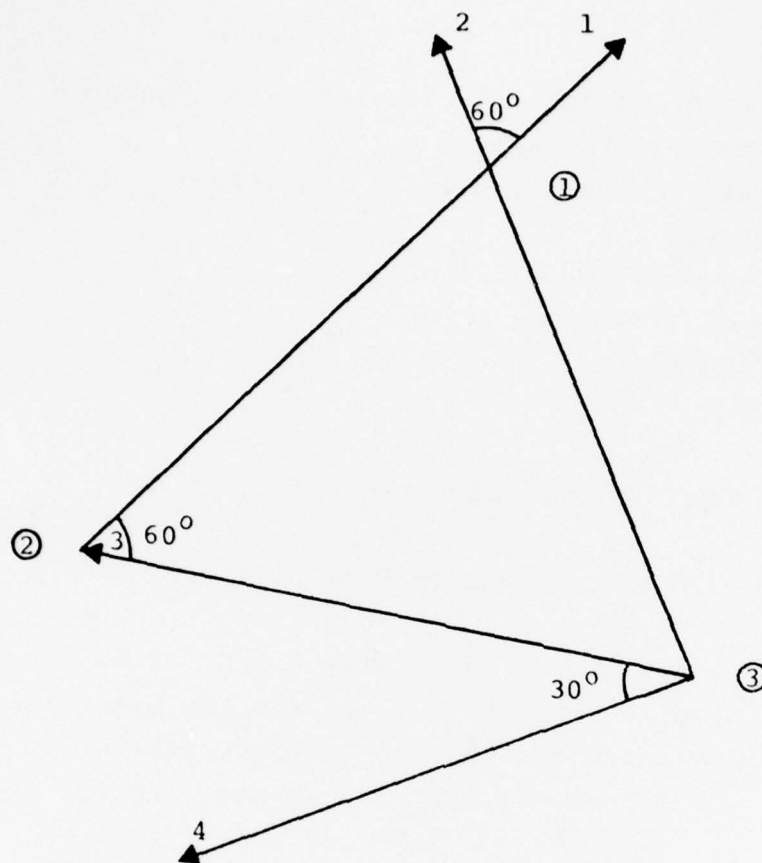
Model detail #		Description			
①		X	1	2	60
②		rightturn	1	3	120
③	{	diverge	2	3	60
	{	diverge	3	4	60

Fig. 5. Descriptions of the airport model of Fig. 1.

```
type
    inttype = (Xjunction, brighty, darky, leftturn, converge,
              diverge, rightturn);
    ltype   = (leftturn, converge, diverge, rightturn);
    intersection = record
                    i:inttype;
                    r1,r2:runway;
                    a:angle;
                  end;
```

Two points regarding these descriptions must be noted here.

1. Angles are measured counterclockwise from r1 to r2.

2. In the Y-junction description, the runway which extends on both the sides of the point of intersection is called the major runway and the runway which is terminated by the point of intersection is called the minor runway. r1 represents the major runway and r2, the minor runway.

The user interface for inputting the model descriptions is still in a stage of development. At the time of writing, the user inputs the model by listing intersections and parallelisms as a text file according to a certain format [6]. Eventually, we plan to automate the model detail computation - intersections, parallelisms & positional relationships - thus requiring that the user only input runways and taxiways as vectors in a coordinate system.

Image Feature Description

The choice of intersections as the principal modeling unit is motivated by the relative ease with which similar image descriptions can be built. In [3], we described techniques for extracting linear features. Apars, or anti-parallel line segments of a certain width, can be used as descriptions of portions of runways. At points in the

94

image where runways intersect or meet, it is possible for the apars to be sharing a common supersegment. A supersegment is a single, piecewise linear, curve providing continuity of edge data in the image. We have called such a collection of apars as saps [2]. A _sap_ (short for super anti-parallel) is a collection of apars with a common supersegment [2] providing one (or both) of the component segments. Moreover, the apars can be ordered according to their occurrence in a single traversal of the supersegment.

It is this data structure of sap that we believe is also useful in providing us a base on which to hypothesize a possible intersection. In Fig. 6, for example, apars 1, 2, 3, and 4 form a sap. It is reasonable to suppose that an intersection is present at the meeting point of the axes of the apars 2 and 3. Although the intersection in the figure is described as an x-junction, the internal description can make a further distinction because apars can be considered as vectors. Thus, a variable of _type_ ltype is used to describe the directions of the apars. While the data sufficient to describe the intersection are the two apars, for gaining ease in subsequent computing, the included angle between them and their directions are both noted.:

```
type   imageint = record
                    t: ltype;
                    a1, a2: dapar;
                    a: angle;
                  end;
```

The _type_ dapar serves to treat apars as vectors. Thus, image descriptions are solely made up of a list of 4-tuples. Further, we order the apars in such a way that the angle is measured anticlockwise from the first apar in the tuple to the second. [At the time of forming these descriptions, it is useful to note the collinearities, viz, apars 1 and 2 are collinear as are 3 and 4].

95
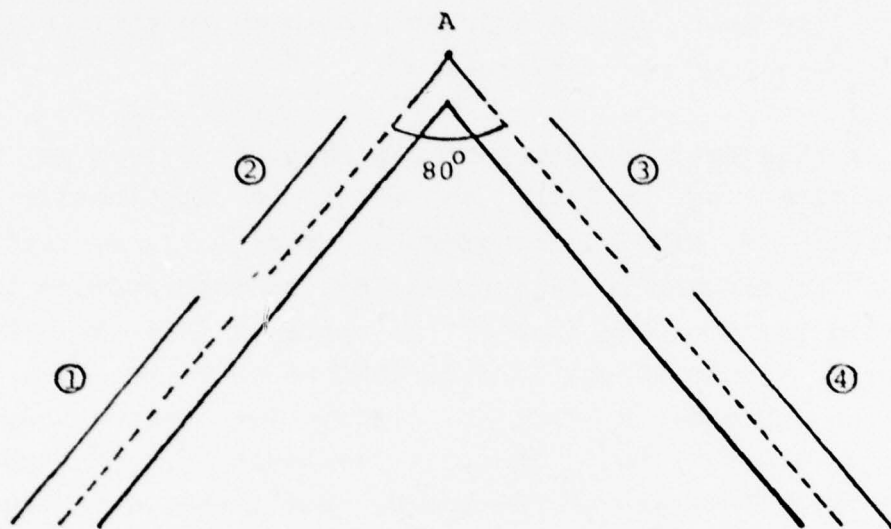
description of } :   X   2   3   80
intersection

Fig. 6 A (typical) sap, and hypothesizing of an intersection

96

The Matching Process

We have viewed the matching process as the problem of relating each model runway to a set of collinear apars. In the algorithms, however, a set of collinear apars can be identified by a representative apar of the set and thus, the matching problem becomes a labeling problem [1,4].

The Labeling Problem. Given a set of units $U = \{u_1, u_2, \ldots, u_m\}$, a set of labels $L = \{\ell_1, \ell_2, \ldots, \ell_n\}$ and a set of constraints as elements of $\{UxL\}^i$, $i=1,2,\ldots,m$ find elements in $\{UxL\}^m$ consistent with the constraints. These elements are called consistent labelings [4] or global constraints [1]. A labeling is a mapping of all the units into single labels.

In [1], Shapiro and Haralick attempt to produce a labeling which maps each unit to a label, where any subset of units and their labels is consistent with some constraint(s) in $\{UxL\}^i$, $i=1,2,\ldots,m$. On the other hand, Freuder [4] considers the problem as one of producing or synthesizing a constraint involving all the units and their labels. These two approaches result in different algorithms: [1] merely reduces the amount of search in an otherwise straightforward search while [4] systematically builds constraints of the next higher order until, finally, a global constraint, or a constraint where all the units are labeled, is found.

In our case, the units are runways, the labels are directed apars and the constraints are the primitive labelings. A primitive labeling is any independent constraint on a (set of) model runway(s). In our problem, however, we do not use unary constraints (involving only one runway), because these unary constraints -- viz., length, width, brightness -- do not seem to provide useful discrimination. Thus, all we have as source of independent constraints are intersections, parallelisms and positional relationships, and these can only form

97

binary constraints (Observe that we represent model intersections involving more than two runways as a set of model intersections involving exactly two runways). A _match_ between a model intersection and an image intersection is achieved when, in an imaginary positioning of the image intersection on the model intersection with one point of intersection on top of the other, the runways and apars are aligned. It is clear that the included angle of the image intersection will be approximately the same as that of the model intersection. At this point, the runways and the corresponding apars may be of the same orientation or of opposite orientation. Thus, in a _primitive labeling_ described below, r1 matches with a1 and r2 matches with a2. Again, the _type_ dapar serves to preserve the orientation information about the apars.:

```
type primitive labeling = record
                r1,r2: runway;
                a1,a2: dapar;
        end;
```

The distinctions mentioned earlier among the various categories of L-junctions and Y-junctions are valuable in the formation of primitive labelings. In general, an image intersection can match an X-type intersection in any one of the four angular positions. With a Y-type model intersection, however, there are only two different matches possible. Finally, with the L-type model intersection there can be only one match. Thus, according to whether the model intersection is X-type, Y-type, or L-type, a maximum of 4, 2, or 1 primitive labelings are formed, respectively, for each matching image-model intersection pair. In all these cases, the apars may match the runways as they are, or with their directions reversed.

The output of our matching is a set of _maximal labelings_. A _maximal labeling_ is a mapping of units into labels where not all units will be mapped; morever, it is not possible to to find another

labeling which is such that -- the set of units having labels in this labeling is a super set of that of the maximal labeling and the corresponding labels are equivalent. The equivalence, in our case, translates to collinearity.

With the problem defined as above, we note here the differences between our algorithm and those in [1] and [4]:

1. In [1], the aim is to generate a labeling which labels all the units; we cannot afford to take such a viewpoint because of the presence of noise in real images.

2. Both in [1] and [4], any constraint not compatible with any primitive or partial labeling is removed from further consideration; again, we cannot afford to remove constraints because we are never sure that a partial labeling is "more correct" than the primitive labeling.

Our search procedure is designed to incorporate the abovementioned differences, and proceeds according to the tree of Fig. 7. Each node of the tree records information about the (partial) labeling formed upto that point in the search and a set of runways responsible for making the labeling. (These runways are sometimes referred to as "runways of a node" in the following text).

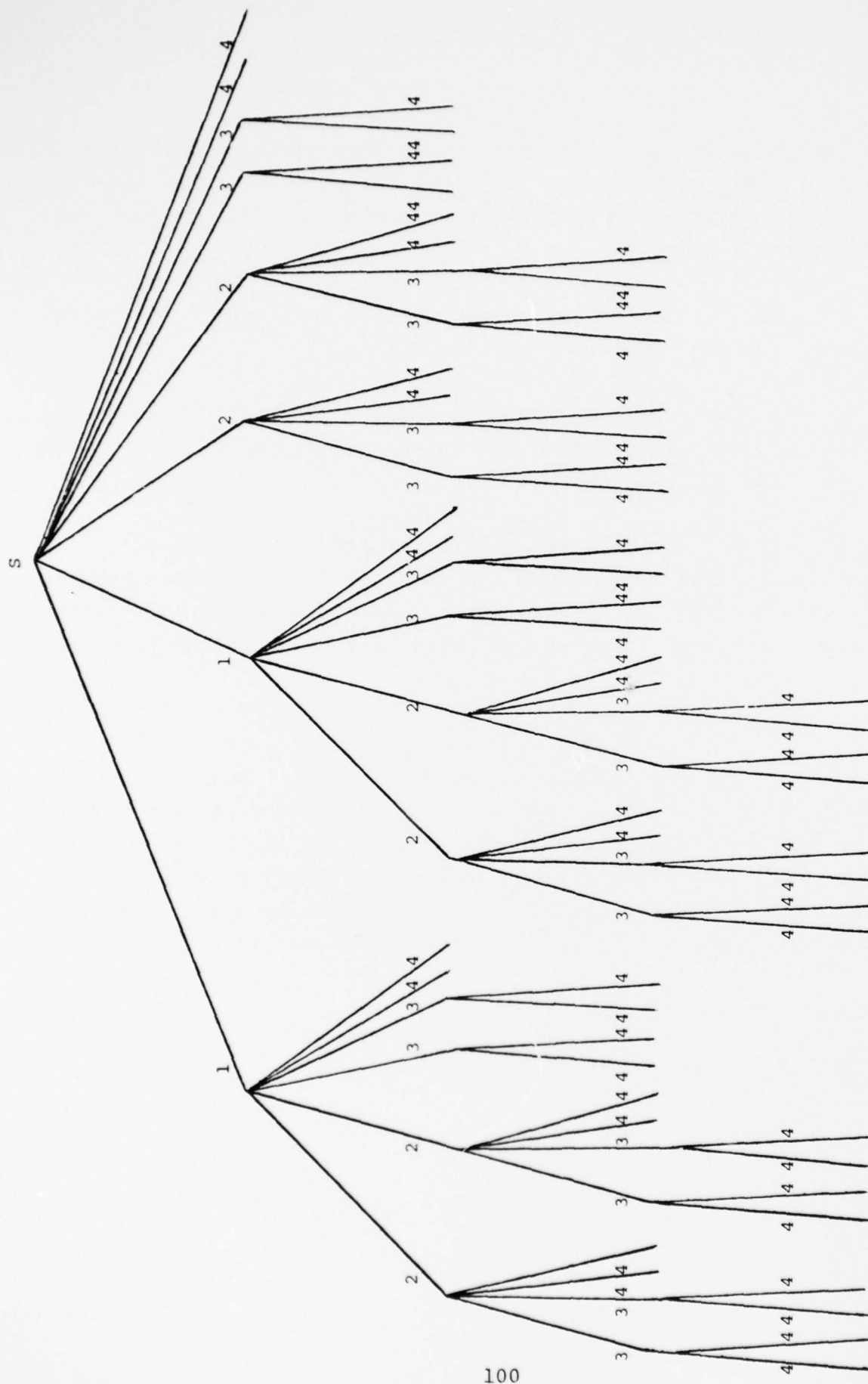Fig. 7   Search tree to find all consistent labelings

```
type    labeling = record
                    a: array[runway] of dapar;
                  end;

node =      record
              l: labeling;
              rused: set of runway;
            end;
```

In the search tree of Fig. 7, we have considered a hypothetical case
where each runway forms exactly two primitive labelings. The numbers
alongside each branch refer to the runway that is used in choosing a
constraint for continuing the search along that branch. While, in
general, this constraint can involve one or more runways, our problem
does not contain unary constraints. These constraints are nothing but
the primitive labelings of a model intersection involving the runway
in question. These primitive labelings are each responsible for the
formation of a successor node.

The initial node S will have an "empty" labeling and an empty set
of runways. Search proceeds by considering any runway that is not in
the node; we consider all the model intersections formed by that
runway and form primitive labelings; each one of the primitive
labelings will enable us to form a successor node. Observe that the
search tree will not be balanced.

The formation of a successor node involves (i) assigning a new
(partial) labeling and (ii) adding the runway responsible for the
formation into the set of runways of the node. The successor node is
formed in such a way that -

1. All the runways that are labeled in either the labeling of
the parent node or the primitive labeling are labeled in the labeling
of the successor node and

101

2. If a runway is labeled both in the labeling of the parent node and in the primitive labeling, then the labels, i.e., apars, must either be same or collinear; further in the new labeling, the runway gets labeled as one of the apars, which is thereafter considered the representative of the collinear group of apars labeling the runway.

3. The runways of the successor node = <u>union</u> of runways of the parent node and the runway responsible for expansion.

Note also that the parent node should also be considered for further search but with a specific change; the runway responsible for the current expansion of the node should be included in its set of runways.   The successor nodes are tested against a global criterion and discarded if they fail to satisfy it. Currently, as implemented now, the global criterion is relative angular disposition of the runways.

When a node can no longer be expanded, it represents a maximal labeling for our problem. However this labeling may not completely characterise the runways because all the runways might not be labeled and further, the labeled runways might not be completely described by the labeling apars. A further search is required to complete the runway descriptions. At the time of writing, this secondary search is not implemented.

Concluding Remarks

The matching procedure we have described essentially exploits only connectivity relationships and hence, in general, we should expect to find many more labelings than the correct one, if it exists. Noise and symmetries also add to this number. We plan to utilise positional relationships in successfully filtering out some globally unwanted labelings. The effectiveness of the procedures desribed here

need to be evaluated on real images.

References

1. Freuder, E.C., "Synthesizing Constraint Expressions," Comm. ACM, Vol. 21, No. 11, November 1978, pp. 958-966.

2. Babu, K. Ramesh and R. Nevatia, "Use of Linear Features in Road Detection," USCIPI Report 860, University of Southern California, March 1979, pp. 62-68.

3. Nevatia R. and K. Ramesh Babu, "Linear Feature Extraction," Proc. ARPA Image Understanding Workshop, Nov. 1978, pp. 73-78.

4. Shapiro, L.G. and R.M. Haralick, "The Consistent Labeling Problem and Some Applications to Scene Analysis," Proc. of 4th IJCPR, Kyoto, Japan, 1978, pp. 616-619.

5. Nevatia, R. and K.E. Price, "Locating Structures in Aerial Images," Proc. of IJCPR, Kyoto, Japan, 1978, pp. 686-690.

6. Price, Keith E., Private Communication.

2.7 Model to Image Matching-Continued

Keith E. Price

This note reports on the current status in the development of a system for matching models of a scene with images of the scene. More detailed discussions and references to other work have appeared in previous semi-annual reports and a complete description will appear in a forthcoming, separate report.

The system uses both region based and edge based segmentation techniques. The edge based method is used to find prominent linear objects (especially roads), and the region based method extracts the other objects which are easily described as connected regions. These segments, lines and regions, are used as the basic elements in the symbolic description of the image, which is completed by extracting various features of the segments. The user model, which internally is the same form as the image description, is derived through a dialog between the program and the user. The features used relative positions, etc.

The model is matched with the image to determine which image segment corresponds with each model element. At this point several improvements have been made. In aerial images the linear feature extraction system tends to break objects into several pieces. Figure 1 is an aerial view of the Stockton, California area with a major highway running from the top to the bottom. Figure 2 shows the major linear features extracted from this image. This major road has been broken into many small pieces, with several large gaps. Also, many of the portions which appear to be connected are actually, in the current linear feature description, several unconnected elements. Now the system allows for multiple matches with a model element. As always, whether a model element is used once or more than once depends on the

104

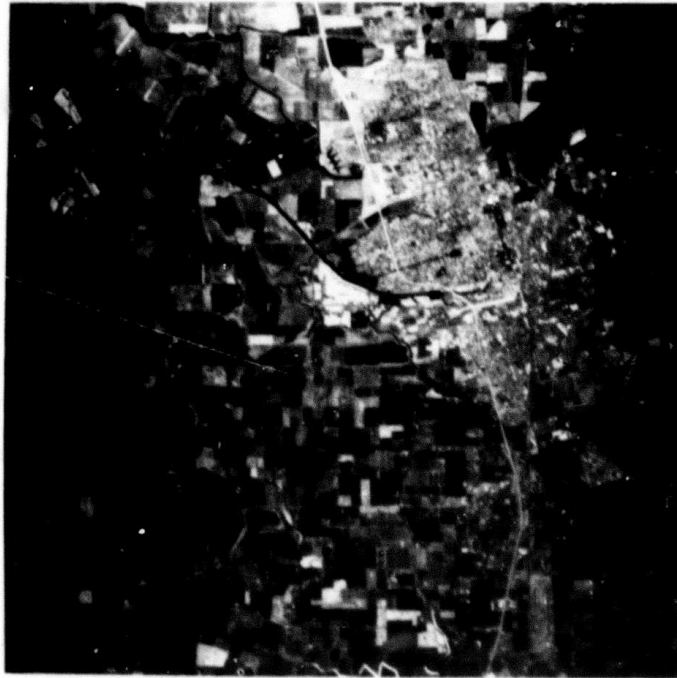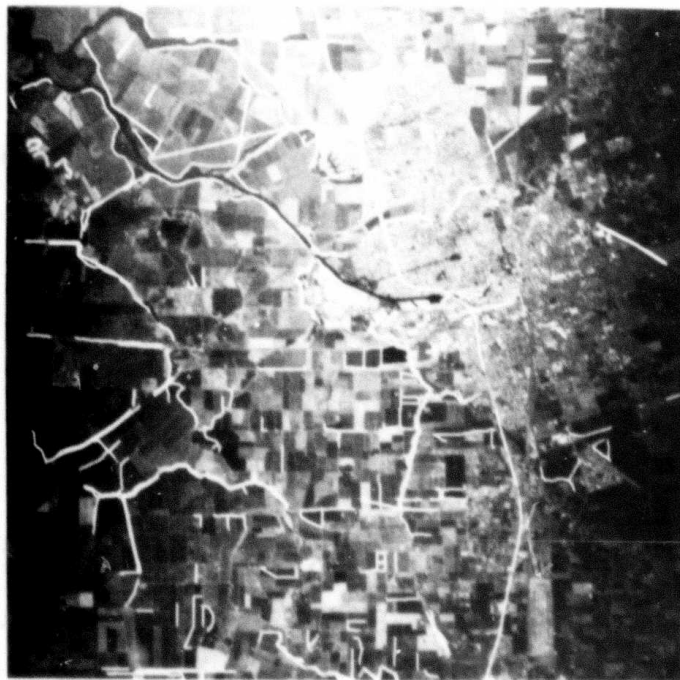Figure 1. Aerial View of Stockton Area

Figure 2. Long Linear Features Located in the
Stockton Image

105

Figure 3. Recognition of Region and Linear Segments in Stockton. This shows the capability of the system to locate multiple pieces of a given object. Note that there are several pairs of channels and roads with overlapping labels

Figure 4. Recognition of Segments in an Aerial View of San Diego

task and model descriptions. Figure 3 shows how this can be used to identify multiple portions of the road segments, with multiple segmentations being given the same name. Also some of the narrow river channels have been located along with the adjacent road, their names overlap because of the automatic label position method. Figure 4 shows another example of multiple segments for a road for a different scene.

2.8 Comparison Between MAP and LMMSE Filters for the Restoration of Images with Poisson Noise

C.M. Lo and A.A. Sawchuk

In several previous reports, mathematical models for images degraded by Poisson noise have been presented and a maximum *a posteriori* (MAP) nonlinear filter for image restoration has been derived [1,2]. In this report, we make a comparison between the linear minimum-mean-square error (LMMSE or Wiener) filter and the MAP filter for restoring images with Poisson noise. The Wiener filter is much easier to implement although it is only a linear processor which is not expected to give as good a result as the nonlinear MAP filter.

Structure of the LMMSE Restoration Filter

From estimation theory [3]-[9] the LMMSE filter transfer function $W(u,v)$ is derived by minimizing the expectation of the squared error between the original object and the estimated object. Use of this criterion or the equivalent orthogonality principle [5]-[8] yields the transfer function of the LMMSE filter as

107

$$W_p(u,v) = \frac{\Phi_{fg}(u,v)}{\Phi_{gg}(u,v)} \tag{1}$$

where Φ_{fg} is the cross-spectral density of the detected image $g(x,y)$ and the object $f(x,y)$ while Φ_{gg} is the spectral density of the detected image $g(x,y)$. From a straight forward substitution into Eq. (1) [10], we have

$$W_p(u,v) = \frac{\bar{N}\mathcal{H}*(u,v)\Phi_f(u,v)}{1+\bar{N}||\mathcal{H}(u,v)||^2\Phi_f(u,v)} \tag{2}$$

where $\bar{N}$ is the mean number of photon counts in the detected image, $\mathcal{H}(u,v)$ is the Fourier transform of the degrading point-spread function (PSF) $h(x,y)$ and $\Phi_f(u,v)$ is the spectral density of the object. The detailed derivation of Eq. (2) is described in [10]. For a linear Gaussian additive noise model, the Wiener filter is [3]-[9]

$$W_W(u,v) = \frac{\mathcal{H}*(u,v)\Phi_f(u,v)}{||\mathcal{H}(u,v)||^2\Phi_f(u,v)+\Phi_n(u,v)} \tag{3}$$

where $\Phi_n(u,v)$ is the spectral density of the noise which is statistically independent of the signal, $\Phi_f(u,v)$ is the spectral density of the object and $\mathcal{H}(u,v)$ is the Fourier transform of $h(x,y)$. Rewriting Eq. (2), the Wiener filter takes the familiar form

$$W(u,v) = \frac{\mathcal{H}*(u,v)}{||\mathcal{H}(u,v)||^2+\frac{1}{\beta}} \tag{4}$$

where

$$\beta \overset{\Delta}{=} \frac{\Phi_f(u,v)}{\Phi_n(u,v)}$$

108

is called the signal-to-noise (SNR) for the linear additive Gaussian noise model.

Referring to Eq. (2) for the Poisson noise model, the LMMSE filter can be written

$$W_p(u,v) = \frac{\aleph^*(u,v)}{||\aleph(u,v)||^2 + \frac{1}{\alpha}}$$

(5)

where $\alpha = \bar{N}\Phi_f(u,v)$. Thus the function $W_p(u,v)$ is the same as $W_W(u,v)$ except that α is defined differently from β. although Poisson noise and its SNR are signal-dependent, α can be called the equivalent signal-to-noise ratio $(SNR)_{eq}$. When the value of is very large, which is the case when the rate function is high, then Eq. (5) becomes

$$W_p(u,v) \simeq \aleph^{-1}(u,v)$$

(6)

and the LMMSE filter approaches the inverse filter in the absence of Poisson noise. Indeed, the larger the rate function, the lesser is the degradation due to Poisson noise. In this case, the LMMSE filter only needs to remove the blurring degradation effects. As discussed previously [1], [2], Poisson noise effects are much more pronounced at low light levels when the value of α is smaller. In this case, the LMMSE filter W_p is dominated by Poisson noise and image signals will be seriously distorted. Thus the performance of the LMMSE filter will be expected to be worse at lower equivalent SNR's. Although the LMMSE filter is based upon the minimum mean-square error criterion, this estimation error is a minimum under the condition that the a priori knowledge is perfect. Functions such as the spectral density of the object Φ_f and the mean number of photon counters $\bar{N}$ must be perfectly

known. In reality, Φ_f and $\bar{N}$ are never perfectly known and must be estimated from the observation [11]. Hence, the actual LMMSE error does not reach the minimum. In short, the LMMSE filter tries to force the solution toward the inverse solution with some sort of smoothness controlled by the equivalent SNR .

Structure of the MAP Restoration Filter

It has been noted earlier [1], [2] that the fundamental MAP estimate contains maximum likelihood (ML) and a priori terms in its solution. The MAP estimate can be written in the equivalent form

$$\hat{f}_{MAP} = \bar{f} + \lambda R_f H^T (q-1) \tag{7}$$

where $\bar{f}$ is the nonstationary mean of the object estimated from the noisy data. The physical interpretation of the MAP estimate is that maximizing the probability $p(\underline{d}|\underline{f})$ forces the solution toward the inverse solution which is the maximum likelihood solution, while maximizing the probability $p(\underline{f})$ is equivalent to enforcing a smoothness criterion. Thus, the MAP filter tries to balance the inverse solution with a smoothness constraint [12]. Another physical interpretation from Eq. (7) is that the MAP estimate tries to move the solution of the estimate $\hat{f}_{MAP}$ from the a priori nonstationary mean $\bar{f}$ to a maximum likelihood solution $\bar{f}_{ML}$.

Experimental Results

Our experimental implementation of the LMMSE filter is based on Eq. (5) using a fast Fourier transform algorithm. The ensemble mean photon counts $\bar{N}$ and the object spectral density Φ_f are estimated from observed photon count data. The estimate of Φ_f can be made by substituting a similar "prototype" spectral density suggested by Cannon in the blind deconvolution process [13], or estimated by an iteration method suggested by Lim in a method of image restoration called spectral subtraction (SSIR) [11]. unfortunately, they assume

the noise is linear signal-independent additive. Because the Poisson noise is signal-dependent, the estimation of ϕ_f is different.

The spectral density of the object ϕ_f is related to the spectral density of the detected image by

$$\phi_d(u,v) = \bar{N} + (\bar{N})^2 |\,|\mathscr{H}(u,v)|\,|^2 \phi_f(u,v) \tag{8}$$

where $\bar{N}$ is ensemble mean number of photon counts and $\mathscr{H}(u,v)$ is the Fourier transform of the PSF $h(x,y)$. Thus, the spectral density of the object ϕ_f can be estimated by an iterative method, although this involves an inverse filter with $\mathscr{H}$. We do not investigate the estimation of the spectral density ϕ_f. Instead we assume that ϕ_f is a white spectral density. The white object spectral density assumption extracts more higher spatial frequency content of many images falls rapidly at high spatial frequencies.

For generality, a two-dimensional moving average blurring point spread function (PSF) is chosen for the simulation rather than a Gaussian blurring PSF because it has singularities and phase reversals in the frequency response. Restored images with the LMMSE filter for different equivalent SNR's are illustrated in Fig. 1. The amount of restoration is controlled by the $\hat{\alpha}$. The serious effects of ill-conditioning can be seen in Fig. 1. The restored image gradually blows up as $\hat{\alpha}$ goes to higher values, while it becomes more noisy as $\hat{\alpha}$ goes to lower values. because the Poisson noise degradation is very pronounced for small values of $\hat{\alpha}$ and the ill-conditioning of the inverse filter is worse at larger values of $\hat{\alpha}$, implementation of the LMMSE filter is very sensitive to the equivalent SNR $\hat{\alpha}$.

Figures 2 and 3 show results with the nonlinear sectioned MAP filter implemented with Newton-Raphson iterative techniques as before. Figure 2 has results with no blur, and Fig. 3 shows results with linear blur. The experiments show that the MAP estimate is superior at all signal-to-noise ratios tested.
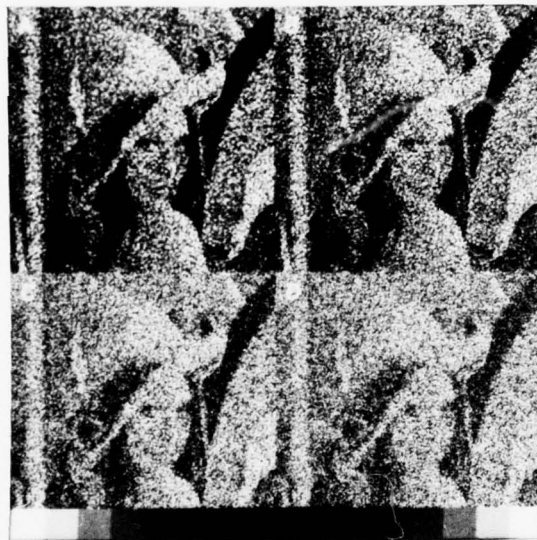
111

Figure 1. Restored images with the LMMSE filter for different
estimated $(SNR)_{eq}$ $\hat{\alpha}$ and 3x3 moving window blur.

(A) Restored image with the LMMSE filter for $\hat{\alpha}=10$
(B) Restored image with the LMMSE filter for $\hat{\alpha}=20$
(C) Restored image with the LMMSE filter for $\hat{\alpha}=40$
(D) Restored image with the LMMSE filter for $\hat{\alpha}=100$

Figure 2a. Images restored by the LMMSE filter and the MAP filter at
$\overline{(SNR)}_{rms}$ = 2.5 with no blurring

    (A)   Original object image
    (B)   Poisson noisy image
    (C)   Image restored by the LMMSE filter
    (D)   Image restored by the MAP filter

113

Figure 2b. Images restored by the LMMSE filter and the MAP filter at $(\overline{SNR})_{rms} = \sqrt{5}$ with no blurring

(A) Original object image
(B) Poisson noisy image
(C) Restored image by the LMMSE filter
(D) Restored image by the MAP filter

Figure 3a. Images restored by the LMMSE filter and the MAP filter
with two-dimensional 3x3 moving window blurring degradation
at $\overline{(SNR)}_{rms} = \sqrt{2.5}$

(A) Original object image
(B) Degraded image
(C) Restored image by the LMMSE filter
(D) Restored image by the MAP filter

Figure 3b. Images restored by the LMMSE filter and the MAP filter
with two-dimensional 3x3 moving window blurring degradation
at $\overline{(SNR)}_{rms} = .5$

(A) Original object image
(B) Degraded image
(C) Restored image by the LMMSE filter
(D) Restored image by the MAP filter

The LMMSE filter and the MAP filter are performing similar functions of balancing the inverse solution with a smoothness constraint. The LMMSE filter uses the equivalent SNR to control the balancing, while the MAP filter uses the covariance matrix R of the object as a measure of the confidence in the nonstationary mean $\underline{f}$ and the maximum likelihood solution $\underline{f}_{ML}$ as a restoration solution. The LMMSE filter is based on the assumption that the detected image intensity can be approximately modeled by a stationary random field. For typical images however, each part of the image generally differs sufficiently from other parts so that the stationarity is not generally valid. Moreover, the LMMSE filtering process is insensitive to abrupt changes of image signals. This results in edge smoothing and contrast reduction. The MAP filter does consider the nonstationary properties of random image fields because it contains an ML term and an a priori term. The MAP filter theoretically needs more a priori knowledge, but in actual implementation, the MAP filter uses less a priori knowledge than the LMMSE filter. Both filters require knowledge of the blurring matrix H and the mean number requires the spectral density of the object Φ_f. It can be concluded that the MAP filter should perform better than the LMMSE filter for the Poisson noise model.

Conclusions

A comparison has been made between the LMMSE filter and the MAP filter with the Poisson noise model. It has been shown that the quality of the restored image of the MAP filter is superior to that of the LMMSE filter by simple subjective evaluation. Boulter [14] has shown that even with large amounts of blurring present in a detected image, a low noise level permits almost complete restoration. However, photon resolved image signals have a very low signal-to-noise ration, making it impossible to obtain perfect restoration for image signals suffering from both blurring and Poisson noise.

117

References

1. C.M. Lo and A.A. Sawchuk, "Estimation of Image Signals with Poisson Noise-I," Semiannual Technical Report, USCIPI 840, Sept. 30, 1978, pp. 135-156.

2. C.M. Lo and A.A. Sawchuk, "Estimation of Blurred Image Signals with Poisson Noise," Semiannual Technical Report, USCIPI 860, March 31, 1979, pp. 90-105.

3. W.K. Pratt, Digital Image Processing, Wiley-Interscience, New York, 1978.

4. H.C. Andrews and B.R. Hunt, Digital Image Restoration, Prentice-Hall, Englewood Cliffs, New Jersey, 1977.

5. H.W. Van Trees, Detection, Estimation, and Modulation Theory, Wiley, New York, 1968.

6. R. Deutsch, Estimation Theory, Prentice-Hall, Englewood Cliffs, New Jersey, 1965.

7. N.E. Nahi, Estimation Theory and Applications, Wiley, New York, 1976.

8. A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, New York, 1965.

9. C.W. Helstrom, "Images Restoration by the Method of Least Squares," J. Opt. Soc. Amer., Vol. 57, pp. 297-303, 1967.

10. J.W. Goodman and J.F. Belsher, "Fundamental Limitations in Linear Invariant Restoration of Atmospherically Degraded Images," Proc. of SPIE, Vol. 75, pp. 141-154, 1976.

11. J.S. Lim, Image Restoration by Short Space Spectral

Subtraction," Report JA 4943, MIT Lincoln Laboratory, Lexington, Massachusetts, February 1979.

12. H.J. Trussell and B.R. Hunt, "Notes on Linear Image Restoration by Maximizing the A Posteriori Probability," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-26, pp. 157-164, 1968.

13. M. Cannon, "Blind Deconvolution of Spatially Invariant Image Blurs with Phase," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-24, pp. 58-63, 1976.

14. J.F. Boulter, "Use of Two-Dimensional Digital Fourier Transforms for Image Processing and Analysis," DREV REPORT R-4025/75, Department of National Defense, Canada, July 1975.

3. Smart Sensor Projects

3.1 DEVELOPMENT OF CUSTOM-DESIGNED INTEGRATED CIRCUITS FOR IMAGE UNDERSTANDING

G.R. Nudd, S.D. Fouse, and T.A. Nussmeier
Hughes Research Laboratories
Malibu, California 90265

and

P.A. Nygaard
Carlsbad Research Center
Carlsbad, CA 92008

ABSTRACT:

This paper describes our on-going program to develop special-purpose charge-coupled device and metal oxide semiconductor integrated circuits for real-time image processing. This work has emphasized the development of circuits that will perform the front-end, or "low-level," processing functions at data rates in excess of 10^6 pixels/sec. We describe the design and fabrication of a third test chip which will perform two-dimensional processing operations over kernel sizes ranging from 3x3 to 26x26 pixels. Included on this chip are data programmable operations for processing over a 5x5 kernel at real-time television rates. In addition, we describe the test facilities we have designed and built to demonstrate the performance of these circuits and the initial test results.

I. INTRODUCTION

A primary aim of the program has been to demonstrate the feasibility of performing image-understanding algorithms in real time. For our purposes, we define "real time" to be equivalent to high-quality television, 7.5-MHz data rate. Even for relatively simple operations on kernels of 3x3 or 5x5 pixels, this represents a speed increase over conventional general-purpose computers of at least two or three orders of magnitude. To achieve this increased thoughput, we have designed and implemented novel charge-coupled device (CCD) and metal oxide semiconductor (MOS) processing architectures that can be integrated into infrared and video cameras. Based on the work funded on this program, we are currently investigating several military applications that require both the sensor and processor to be integrated onto a single chip (the so-called "smart-sensor" philosophy).

We have designed, built, and tested three integrated-circuit test chips containing the 14 algorithms listed in Table 1. Each chip has been used to demonstrate a different approach to image analysis. The first chip shown in Figure 1 was aimed at demonstrating a novel two-dimensional CCD filtering approach which allows concatenation of several (in this case 5) image-understanding operations and has been designed to be integrated into the sensor directly at the focal plane. Each of the functions on this chip operates over a 3x3 array of picture elements and provides a single processed picture element for each new input pixel. The circuit accepts three lines of video data equivalent to the 3x3 array and as such requires two external analog delay lines when operated from a vidicon or commercial camera, as shown in Figure 2. In our initial

121

Table 1. Integrated Circuit Primitives Developed to Date

9136—1

TEST CHIP NUMBERS	ALGORITHMS IMPLEMENTED	KERNEL SIZE	OPERATIONS PER PIXEL	PERFORMANCE			STATUS
				NUMBER OF BITS	SPEED (PIXEL RATE)	EFFECTIVE OPERATION RATE	
I	EDGE DETECTION	3 × 3	16	4	5 kHz	90 KOPS	DEVELOPED AND TESTED
	HIGH-PASS SPATIAL FILTER	3 × 3	18	4	5 kHz	90 KOPS	
	LAPLACIAN	3 × 3	13	4	5 kHz	65 KOPS	
	12 dB/APERTURE CORRECTOR	3 × 3	18	4	5 kHz	90 KOPS	
II	SOBEL	3 × 3	16	4	2 MHz	32 MOPS	DEVELOPED AND TESTED
	MEAN	3 × 3	9	4	2 MHz	18 MOPS	
	UNSHARP MASKING	3 × 3	13	4	2 MHz	26 MOPS	
	BINARIZATION	3 × 3	10	4	2 MHz	20 MOPS	
	ADAPTIVE STRETCH	3 × 3	12	4	2 MHz	24 MOPS	
III	LAPLACIAN	3 × 3	13	6[+]	7 MHz[+]	91 MOPS[+]	IN PROCESS
	MASK PROGRAMMABLE	7 × 7	98	6	7 MHz	636 MOPS	
	PROGRAMMABLE	5 × 5	50	6	7 MHz	350 MOPS	
	PLUS SHAPED MEDIAN			6	7 MHz		
	BIPOLAR CONVOLUTION	26 × 26	1352	6	7 MHz	~10^4 MOPS	

[+]PREDICTED PERFORMANCE BASED ON DESIGN

122

TWO DIMENSIONAL
FILTER

OUTPUT
CIRCUITRY
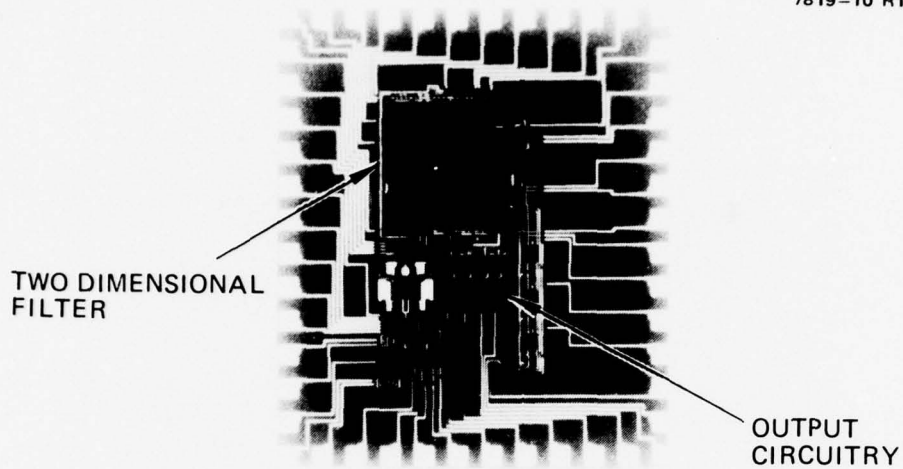
Figure 1.   Photomicrograph of test chip I.

9136–2

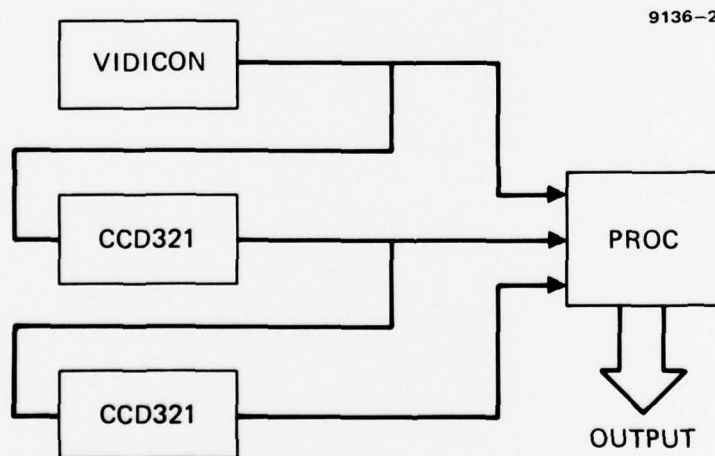VIDICON

CCD321

CCD321

PROC

OUTPUT

Figure 2.   Formation of the 3x3 pixel array using
external analog delay lines.

work to operate these circuits in real time, we have used Fairchild CCD 321 analog line delays. These have worked quite well but limit both the dynamic range and the signal-to-noise ratio of the processor. We are currently investigating techniques for incorporating the circuits directly into a CCD imager as shown in Figure 3.

The second chip, shown in Figure 4, contains five individual circuits again using a 3x3 pixel kernel and is aimed at demonstrating adaptive processing using the local mean as the control. This circuit has been operated directly from both a vidicon and CCD camera with an overall processing accuracy equivalent to 4 bits. The operations performed in addition to the 3x3 average are adaptive stretch, binarization based on the local mean, unsharp masking, and Sobel edge detection. Under a parallel contract with Night Vision Laboratories, Fort Belvoir, Virginia, we have integrated these circuits into a demonstration processor, shown in Figure 5. At the request of the customer, we incorporated a CCD field delay to remove the interlace and provide a processing capability on adjacent lines of video. This processor has been operated at a 4-MHz clock rate, and the results are reported in Ref 1.

Our recent work has been concerned with the design, processing, and initial evaluation of a third test chip, which is aimed at demonstrating processing techniques using larger kernel sizes (as high as 26x26 pixels) and demonstrating a programmable capability. After many problems and delays in obtaining a satisfactory mask set, we have now completed the processing of this chip and are currently collecting preliminary per- formance data on each of the five circuits, as descirbed below.
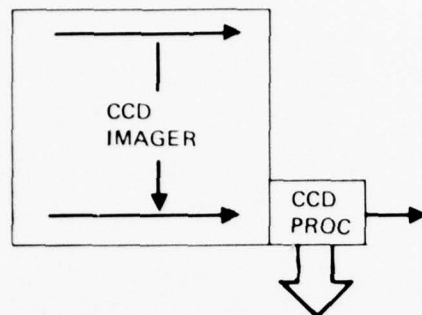
9136-3

Figure 3.   Technique for integrated CCD
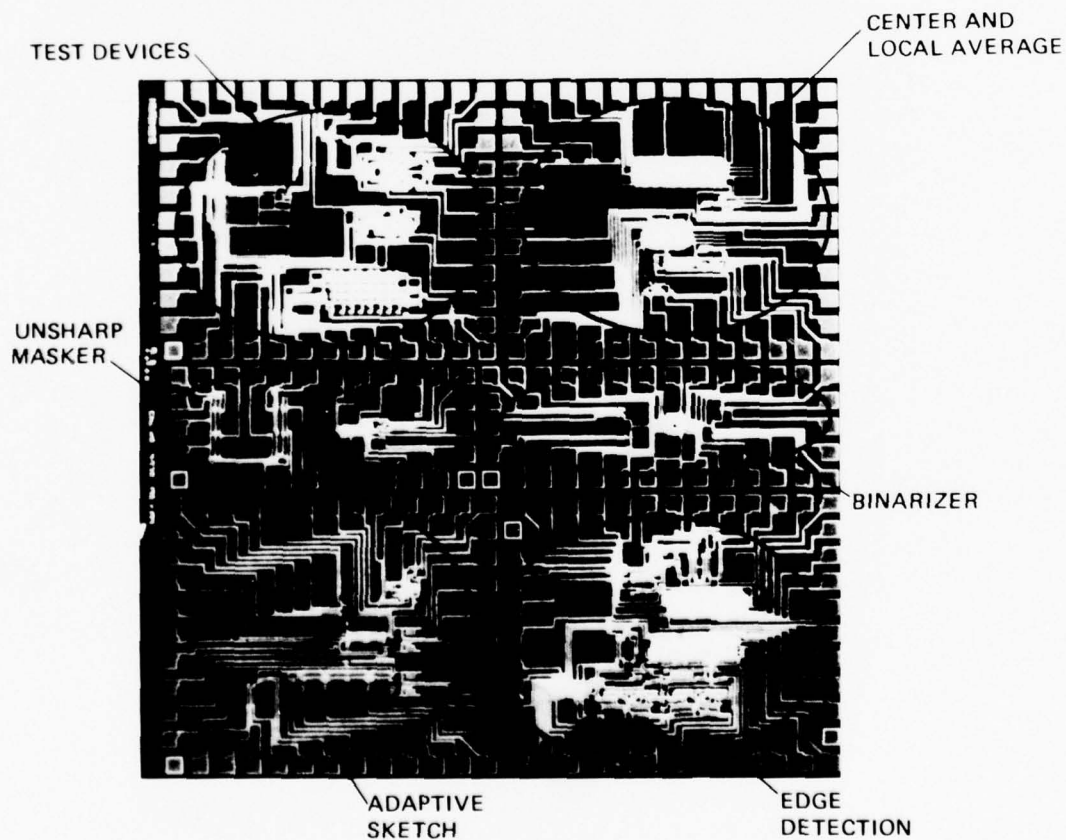images and processor.

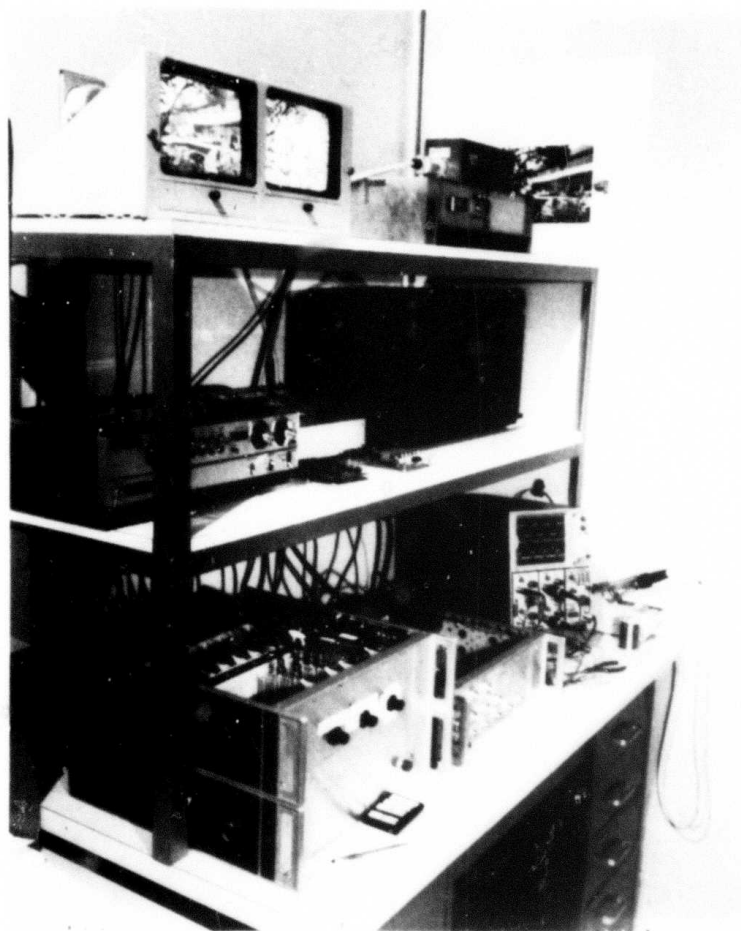
Figure 4.   Photomicrograph of test chip II.

Figure 5.   Real-time test facility for test chip II.

II. PROGRESS ON TEST CHIP III

The principal effort this period has been with the design, simulation,
and processing of this chip. Five functions, a 7x7 mask programmable array,
a 3x3 Laplacian, a median operator, a 5x5 voltage programmable convolution,
and a large 26x26 element convolution for the primal sketch,[2] are included.
The design goal is for a 15-MHz clock rate and an overall processing
accuracy equivalent to 6 bits. The resolution of the circuit lithography
is 5 to 7 μm, equivalent to commercial optical techniques, and the technology
is n-type surface channel. The bandwidth requirements for this chip are
towards the high end of the speed capability range for surface channel
devices and hence represent a considerable challenge. Also, the kernel
size has been considerably extended from the 9 pixels used in our previous
work. The largest processor, the convolution for the primal sketch,
contains 338 pixels. Further, the dynamic range required by the operators
contained on the chip is much increased, representing approximately
8 bits, and we are including special techniques to achieve this.
Probably the most significant challenge we are addressing on this chip is
the development of programmable processing kernels. The concept here is
to develop a general-purpose convolutional processor that can accept
data at real-time video rates and can adapt its kernel size and weights
either in a preprogrammed way or in response to the processed output at
a speed higher than the frame rate (>30 Hz). If such a device can be
developed with accuracy equivalent to 6 bits, it will find very wide-
spread general utility in image analysis and understanding. At present,
the kernel size for this circuit is 5x5, but there is no fundamental
limit preventing this from being significantly increased. To meet the

significantly increased demands both in terms of speed and dynamic range, we have included an on-chip sample and hold to both reduce the output noise and lower the clock feedthrough. This should significantly increase the performance of the functions, particularly at high speed. A schematic of the circuit is shown in Figure 6. This device has been simulated to operate at an 11-MHz data rate and provide a 60-dB common mode rejection driving a 30-pF output load. This circuit is included in each of the CCD functions.

A photomicrograph of the full chip showing each circuit and the test devices is shown in Figure 7. The chip itself is approximately 225 mils^2, which is slightly larger than our previous one (191 mils^2). This has resulted in fewer dice per wafer, thus requiring higher yield to provide acceptable quantitites for testing. We are currently processing 11 wafers, each with 36 dice/wafer. This should hopefully result in enough acceptable circuits for initial testing. Later we will process an additional lot when the chip's initial operating parameters have been determined.

This chip has now been designed and simulated using the circuit analysis and simulation program SPIcE. The individual cells have been drawn and the composite digitized by the mask maker. Because of high demand in the IC market, the turnaround at the mask maker has been some-what longer than originally anticipated. In addition to this delay, the initial mask set received was incomplete; also, some of the masks were reversed field. The net effect of these two delays is an anticipated schedule slippage of several months. We have only recently completed the processing required prior to short testing the devices, dicing the
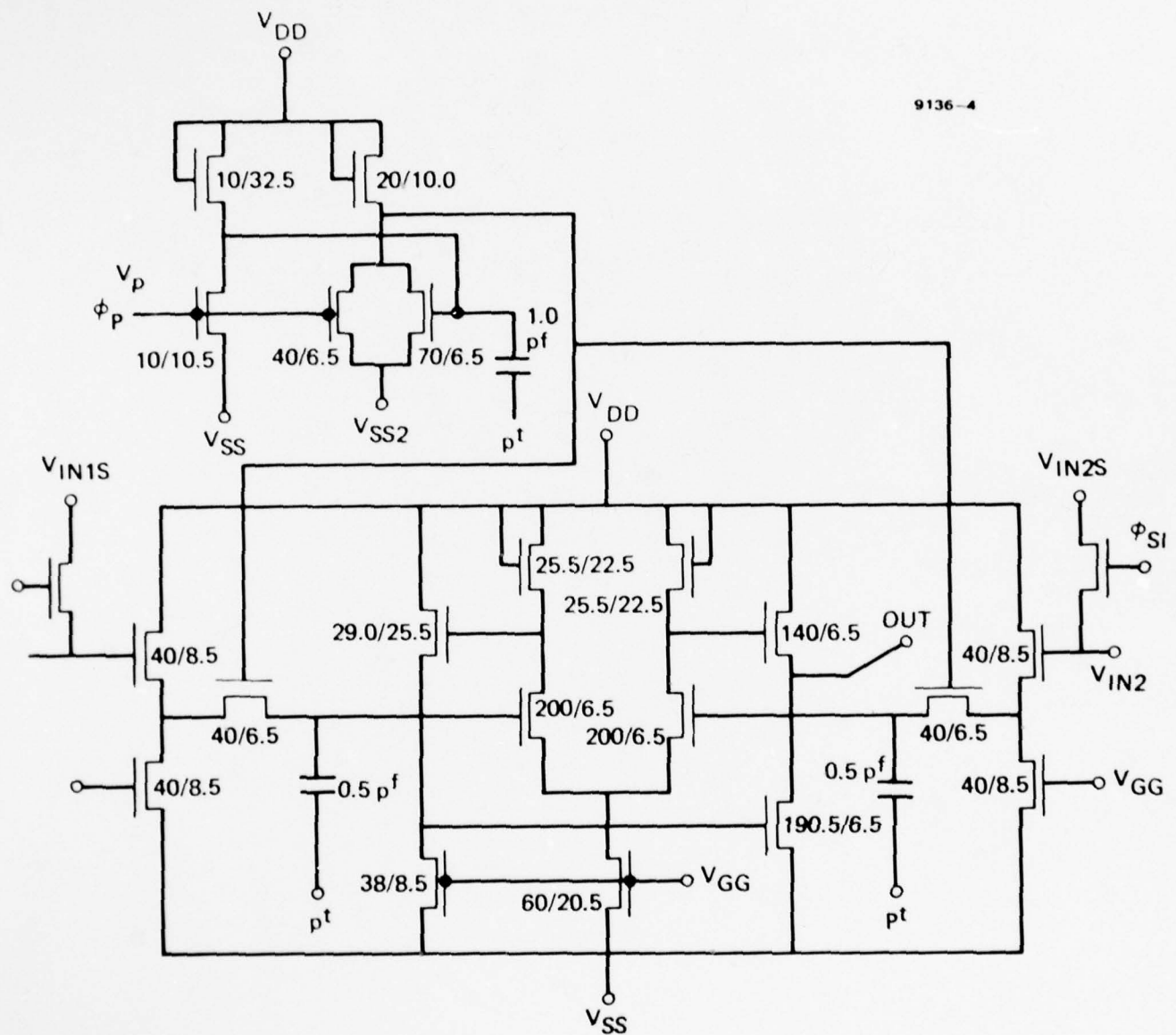
9136-4

Figure 6. Schematic of on-chip sample and hold.
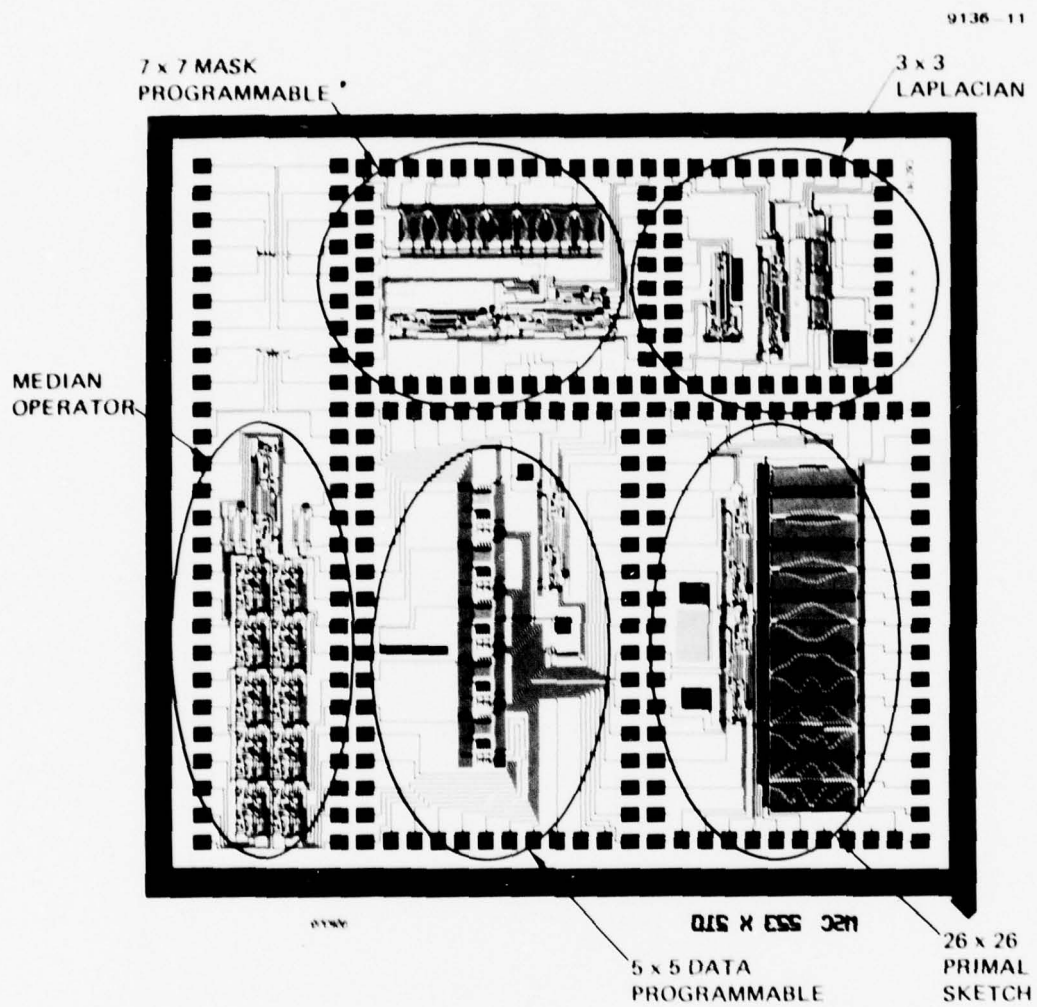
129

Figure 7. Photomicrograph of image understanding chip III.

wafer, and packaging. However, this process is now complete, and we have started the initial performance evaluation. Because of the larger kernel size and the significantly different characteristics of these circuits, we essentially have had to rebuild the test facility. Considerable time and effort was expended on this during this period, as discussed below.

III. PERFORMANCE, EVALUATION, AND TESTING

To be able to effectively test the performance of the microelectronic circuits developed, it has been necessary to develop an appropriate test facility. The essential components of this system are shown in Figure 8. The functions developed on the test chip include both two-phase and single-phase circuits, each requiring a clock swing of at least 20 V. To achieve this, we have developed a driven system that operates from an 8-channel (T^2L voltage level) word generator and uses these outputs to develop MOS level waveforms. This generator is also used to develop the necessary diffusion and reset pulses. The word generator can be clocked at a 20-MHz bit rate to process a 3-MHz bandwidth signal. The imagery will be obtained either from a stored data base or a vidicon camera. In either case, the data requires formatting into several parallel video lines to form the appropriate kernel size. The system to do this has now been completed. It consists of a 10-MHz, 8-bit analog-to-digital converter system that provides input to a 24-kbit RAM register. The RAM register provides a delay equivalent to six horizontal lines. A digital-to-analog converter is included after each 4 kbits to provide the analog output from the adjacent lines. The necessary hardware to provide this facility has now been completed and the system tested with
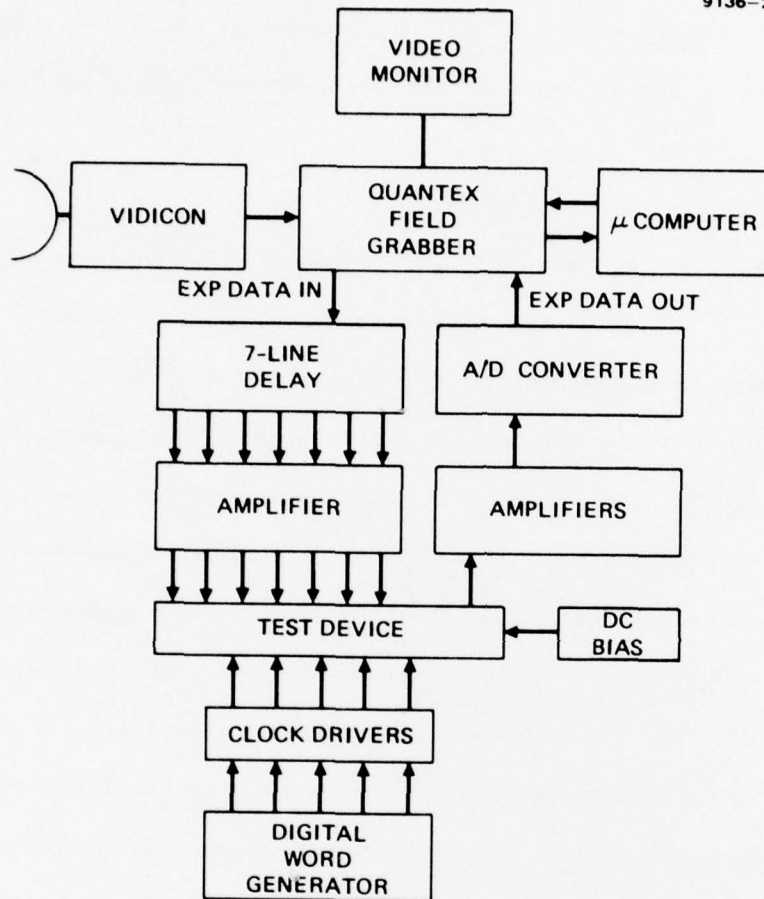
131

Figure 8.   Schematic of test facility.

a commercial vidicon. In addition, a signal conditioner box, which both translates the dc level of the resulting video data and can provide the necessary variable gain, has been designed and built. We also have provided the capability to vary both the spatial and temporal resolution of the processor and have investigated several commercial "frame grabbers" and digital memories to provide this facility. The system we have built is based on the Quantex Field Grabber with resolution of 256x256 pixels each with 6 bits of gray scale. We have interfaced this system to an IMSAI 8080 microcomputer for evaluation and have written several software packages to manipulate the data to provide both simulation of image-understanding operations and manipulation for display purposes.

We are currently using this facility to evaluate the first lot of test wafers. We have decided to investigate the 5x5 data programmable convolution and the median operator initially. The schematic of the 5x5 programmable operator is shown in Figure 9. Essentially, it accepts five parallel lines of video data and performs a 25-point bipolar convolution on a sliding 5x5 pixel array. The mathematical formulation of the processor is given by

$$\hat{I}_{k,1} = \sum_{i=k-2}^{k+3} \sum_{j=1-2}^{1+3} I_{i,j} \cdot W_{ij} \quad ,$$

where $\hat{I}$ is the intensity of the processed image, I is the original image, and the W_{ij} are the programmable weights. The processor consists of a two-dimensional floating gate array with 25 voltage-controlled taps. This array overlays five separate CCD delay lines through which charge equivalent to each picture intensity is clocked. A single source follower
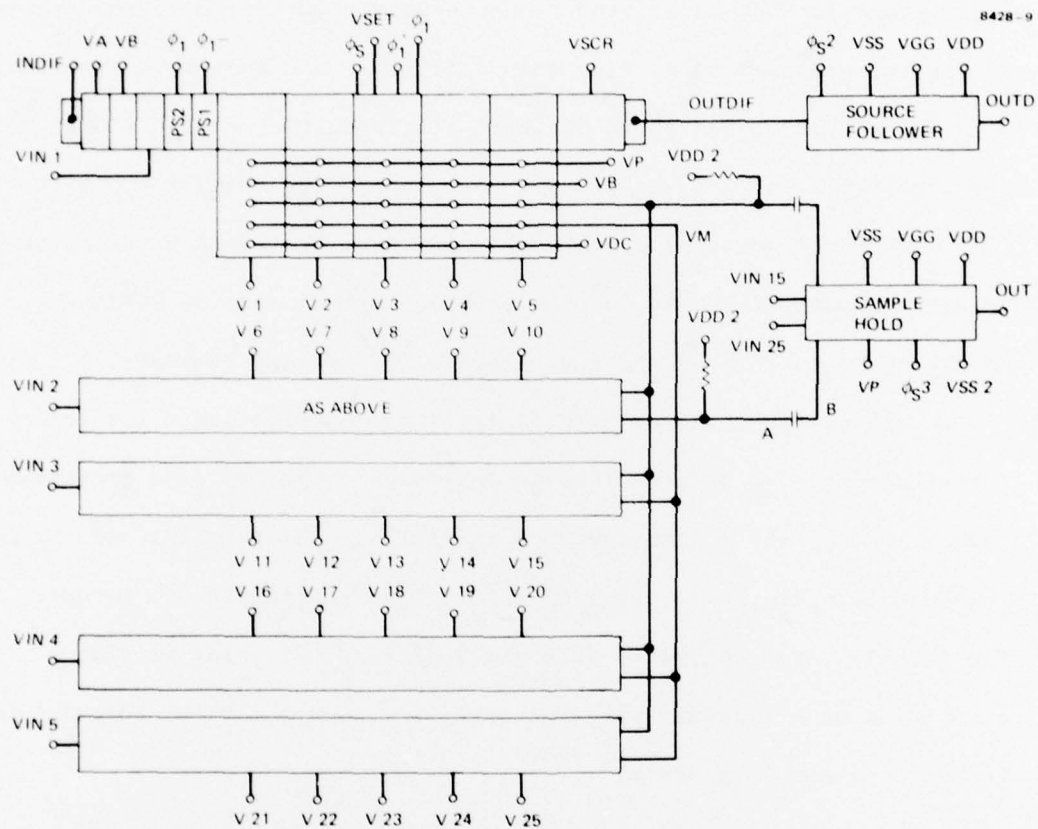
133

Figure 9. Schematic of 5x5 programmable filter.

at the end of all of the delay lines is used to detect the linearity and transfer efficiency of the basic CCD structure. An example of the operation in this mode is given in Figure 10. The input signal is shown on the upper trace, and the output resulting from two cycles of this waveform is shown on the lower trace. The operator here is at ∿6 kHz, equivalent to the pixel rate of the stored data base microprocessor system. The photograph illustrates the need for an output stage including a sample and hold. The reset pulses occurring each pixel interval can be seen to have about the same magnitude as the diffusion output. These are due entirely to feedthrough and are unwanted outputs. To avoid this, we have tested the on-chip sample and hold as shown in Figure 11. In this case, the input signal consists of a single frequency that is sampled each 60 μsec. At these intervals, the output level of the waveform is sampled and frozen until the next sample pulse. This circuit is included on all the active CCD outputs and is used to eliminate the unwanted clock to reset feedthroughs. The initial results obtained from the floating gate are shown in Figure 12(a and b). In each case, the input signal is shown on the upper trace and the resulting impulse response shown below. In Figure 12(a) the voltage settings on the floating arrays are equivalent to 0, -1, 0, -1, 0, and in Figure 12(b) to 0, 1, 0, -1, 0. In each case, the input response equivalent to this setting is obtained, verifying both the programmability and the capability of bipolar weighting. Figure 12(c) shows the impulse response equivalent to 1, 1, 1, 1, 1. Although these are preliminary results, it is clear that all the necessary functions are operating on the chip. The dynamic range of the device obtained so far is illustrated in Figure 13. It is equivalent to about 14 gray levels.
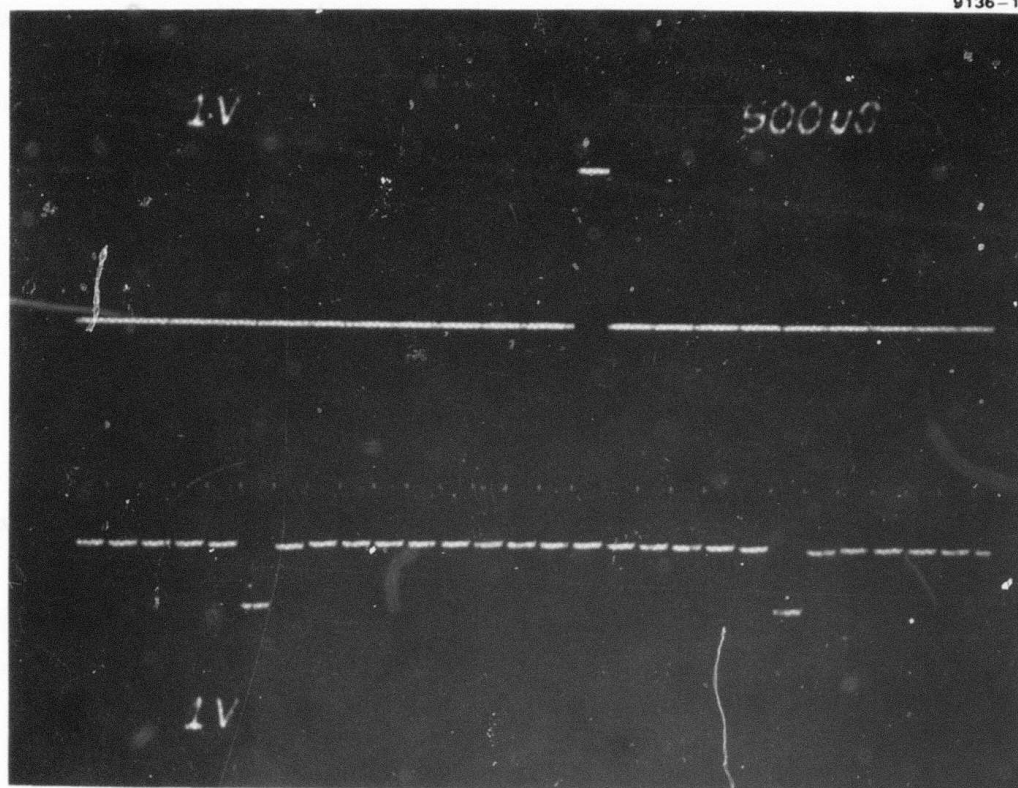
135

Figure 10.   Output from source-follower on the 5x5 programmable processor.
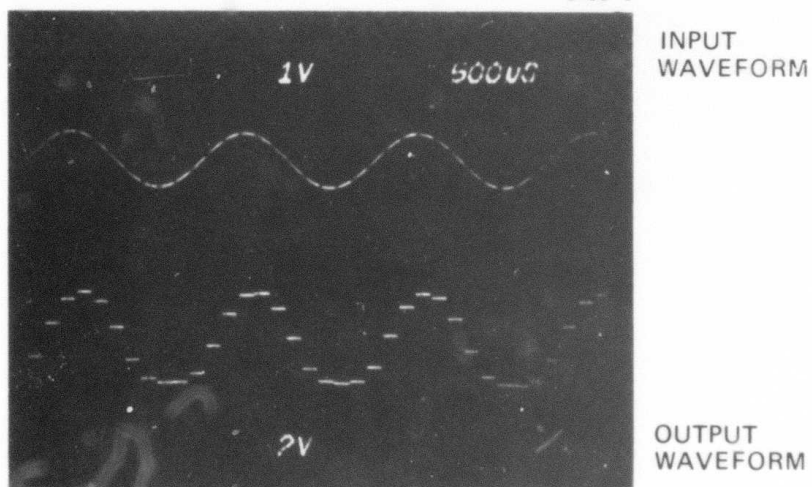
INPUT
WAVEFORM

OUTPUT
WAVEFORM

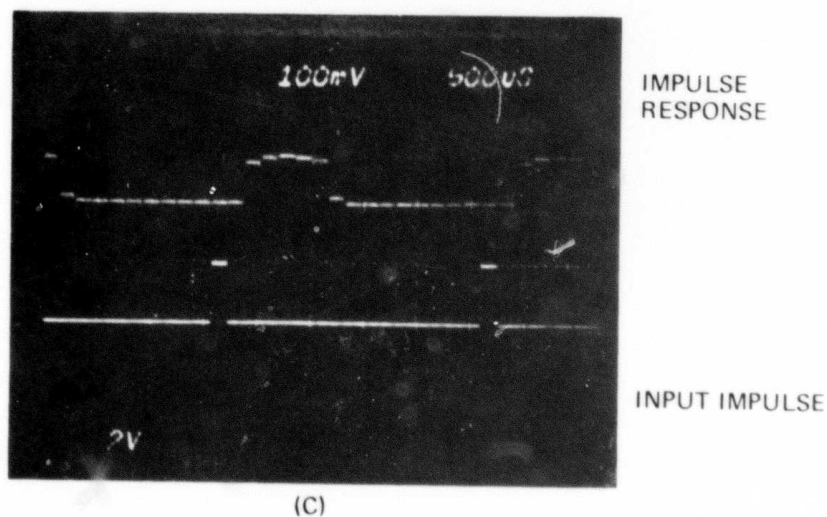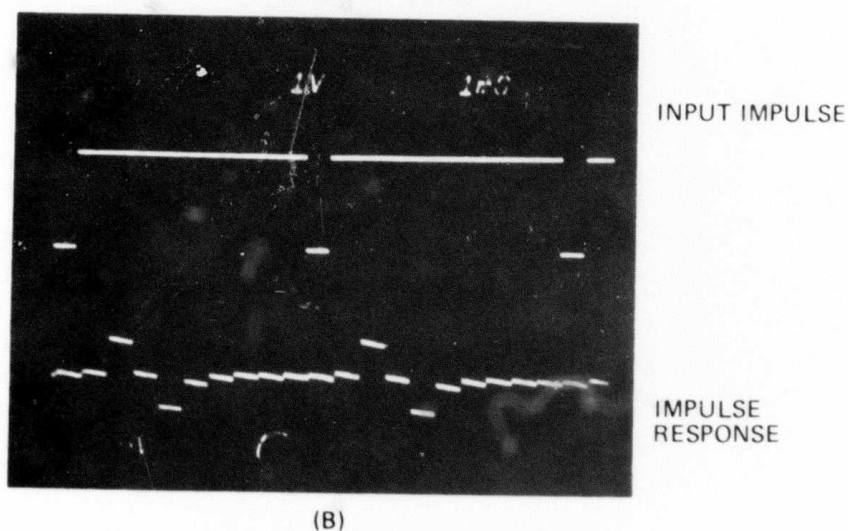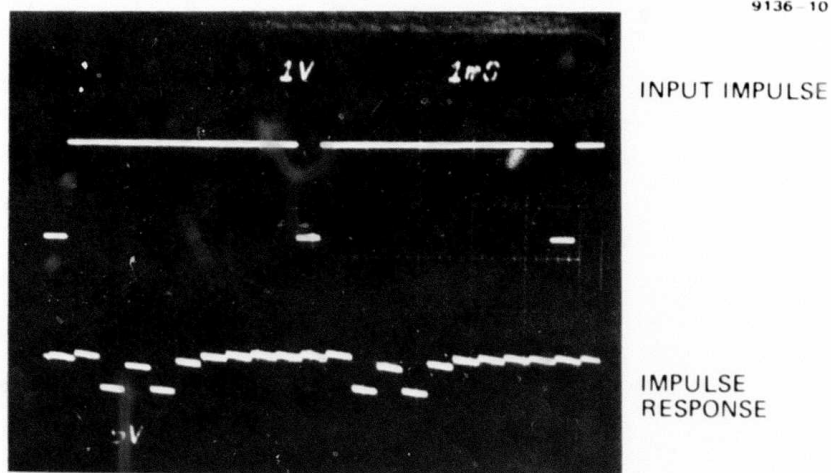Figure 11.   Operation of the on-chip sample and hold.

Figure 12. Impulse response of programmable array for setting
(0, -1, 0, -1, 0), (0, 1, 0, -1, 0), and
(1, 1, 1, 1, 1).

137

9136-9

OUTPUT
VOLTAGE
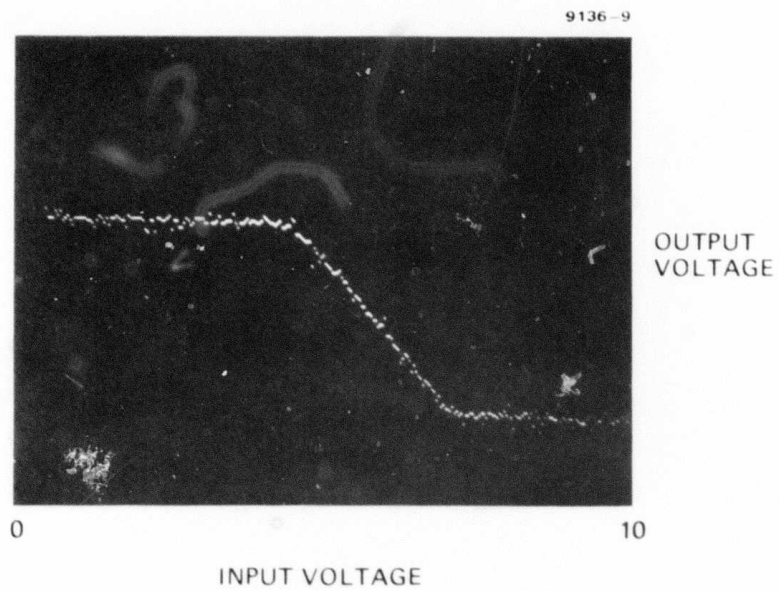
0   10

INPUT VOLTAGE

Figure 13.   Linearity and dynamic range of 5x5 pro-
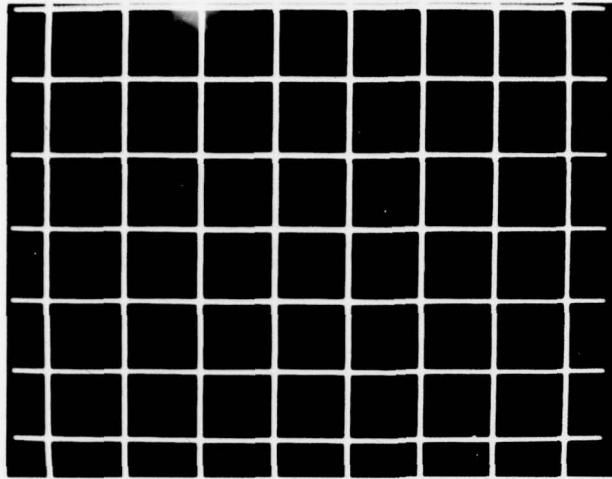   grammable processor.

The granularity on the trace shown is due in part to the microprocessor

system used to obtain this transfer characteristic. Our work will continue

on this circuit to improve both the accuracy and dynamic range and to

increase the speed to 7.5 MHz. We have, however, started to process some

test-patterns, as shown in Figure 14. The first consists of a two-

dimensional grid with lines each a single pixel in width. This has been

processed with the processor set for a 1, 1, 1, 1, 1 impulse response in

both x and y. The resulting grid with each line 5 pixels wide is shown

in Figure 14(b). This corresponds directly to the simulation shown in

Figure 14(c). A similar result for the picture of the house is shown

in Figure 15. The impulse response for the programmable array setup to

be both a plus shape and a cross shape are shown in Figure 16.

During the next phase of the program, we will continue the testing

and evaluation of the circuits and provide a detailed performance

review.

IV. STUDY OF VLSI FOR IMAGE UNDERSTANDING

In addition to the work on the CCD/MOS circuitry and building a

new test facility, we have begun an analysis program to determine the

impact and advantages of very large scale integration (VLSI) on image

understanding. The on-going work in software and algorithm development

clearly is leading to an ever-increasing demand for computational

throughput. Further, it is evident that, even with the most sophisticated

general-purpose machines, the processing times are incompatible with any

real-time application. An apparent solution to these issues is the

development of new processing architectures based on the latest

technologies. Two significant developments have been taking place in

microelectronics in the past several years. The first is the development

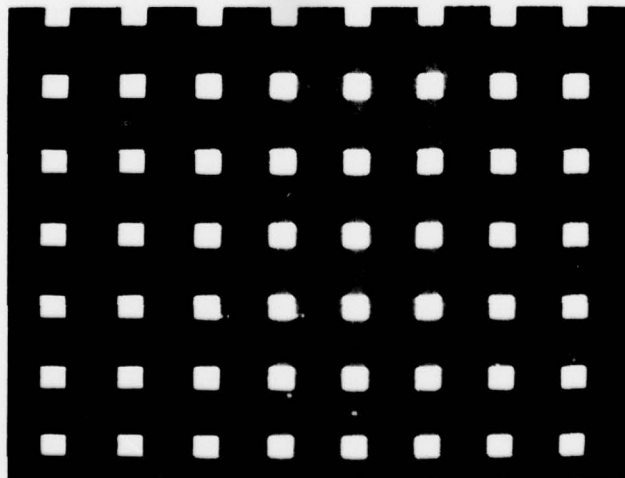of a variety of new technolgoies such as DMOS, CMOS/SOS, I^2L, ECL, and

139

GRID TEST PATTERN

ORIGINAL
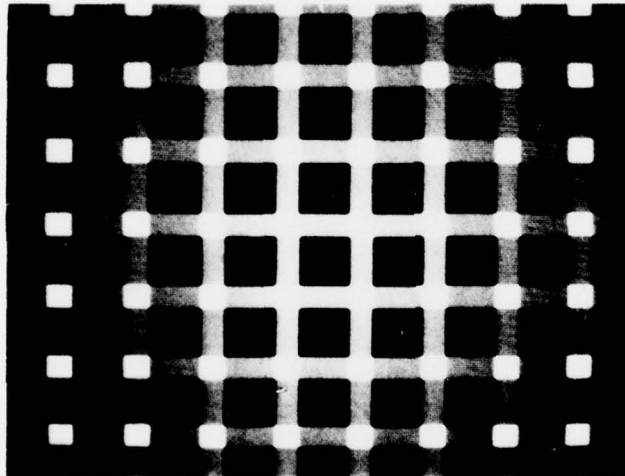
COMPUTER PROCESSED

CHIP PROCESSED
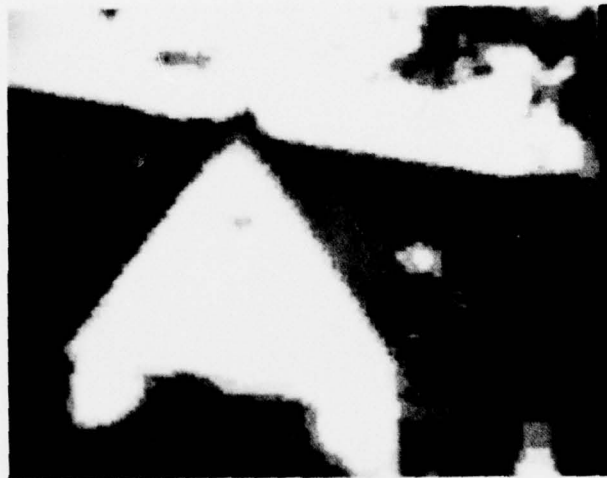
Figure 14. Performance of 5x5 programmable
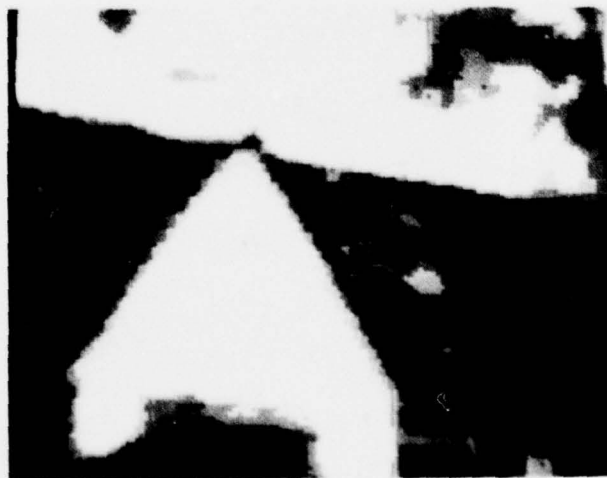processor on test pattern. (all
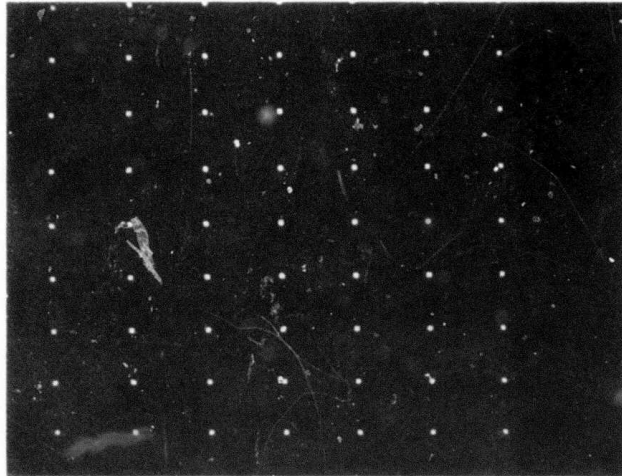weights equivalent to unity).

140

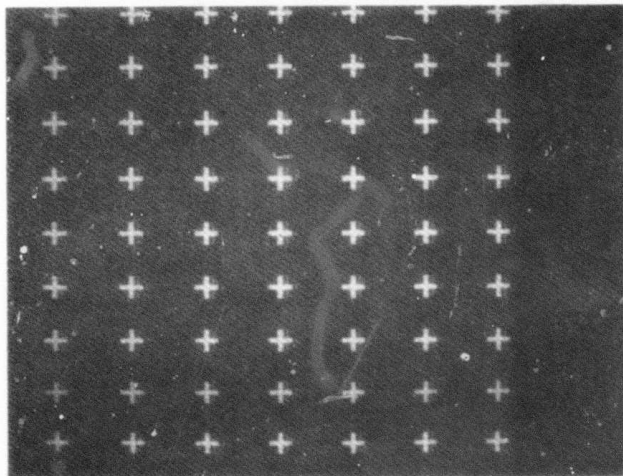ORIGINAL

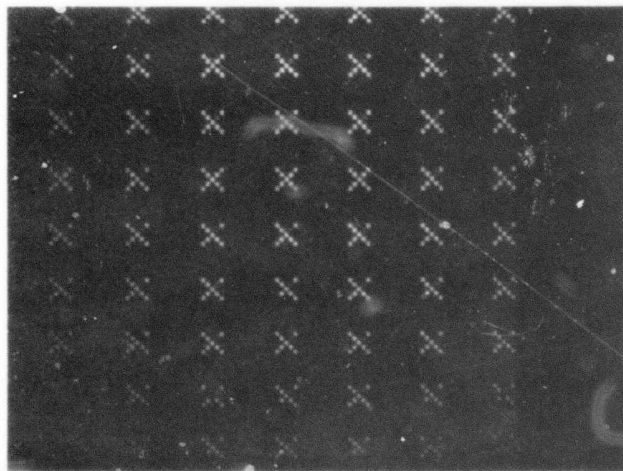COMPUTER PROCESSED

CHIP PROCESSED

Figure 15. Example of processor used on stored
imagery (settings as for Figure 14).

INPUT SIGNAL

PLUS SETTING

CROSS SETTING

Figure 16.   Impulse response of programmable
filter.

GaAs. Each offers a different combination of parameters in terms of

speed, power consumption, and the possible level of integration. It is

of particular importance to the I.U. program to be able to evaluate

the advantages and constraints of all the available technologies with

respect to our programs. In addition, with increased resolution of

integrated-circuit features and decreased power consumption, the level

of integration within each chip has greatly increased. For example,

in highly regular arrays, such as memories, as many as 10^5 to 10^6

functions can be integrated in each chip. This development will also

have profound effects.

To some extent, the design and architecture of high-density

circuitry other than memories and commercial microprocessors have lagged

behind these developments, and a significant advance can be expected

from the optimum design of image-processing architectures. To achieve

the maximum advantages of VLSI for the I.U. problem, two precepts must be

adhered to. First, the designs and architectures must, where possible

provide concurrent operation. The obvious bottlenecks that result in

the von Neuman concepts of a single arithmetic unit, etc. can be cir-

cumvented by highly localized operation with multiple primitives. This

removes many of the problems from the processor and places them in the

control and data distribution system. There is, for example, a need for

localized distributed storage and memory associated with each primitive

for data and instruction queuing. In addition, both for ease of design

and for increased packing density, the circuitry must be highly regular

on the silicon surface.

143

From our previous work on this program, it is clear that, using current state-of-the-art technology, low to intermediate level primitives can be built that will provide real-time operation without requiring extensive area on the silicon. From this work, we anticipate that five or six primitives might be included in a single 200-mil^2 chip. In support of this concept, we are currently investigating concepts for data distribution and intelligent local storage as well as techniques such as residue operation and number theoretic transformation for regularizing the processes themselves. The eventual aim of this work, which will continue in the next period, is to determine an optimum way of mapping the algorithms and processors onto the two-dimensional silicon.

V. SUMMARY AND FUTURE PLANS

During the current period, our work has been concentrated in three areas: the fabrication of Test Chip III, the development of an effective test facility and preliminary testing of the circuits, and an initial study of the effect of VLSI on image understanding. Progress in each of these areas has been satisfactory, although unexpected delays have been encountered primilarly with the outside mask maker. The problems with this vendor have created an unavoidable delay of at least two months and have been largely responsible for our delay in the testing schedule. However, all 11 masks have now been received, and the wafers have been processed and short tested. We have completed nearly all the work on the test facility and are now getting initial test results. This work will continue next period, and we intend to issue an interim report detailing the test results as available. In addition to this, we have started our VLSI study and are starting to formulate an approach for this major task next year.

144

REFERENCES

1. G.R. Nudd, G.D. Thurmond, and S.D. Fouse, "A Charge Coupled
    Device Image Processor for Smart Sensor Applications," S.P.I.E.
    Conference, San Diego, August 30-31, 1978 S.P.I.E. Vol 155
    pp 15-22.

2. D. Marr and T. Poggio, "A Computational Theory of Human Stereo
    Vision," M.I.T. memo 451 (1977).